THE AI TOOLCHAIN

NO. 062

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 14, 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 291 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 49 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. Kiro now sandboxes agent shell commands locally and can auto-approve them inside that boundary, so ended runs stop being a trust decision per command. LiteLLM's

vim ? help

Auto-Router classifies request complexity and sends easy calls to cheap models. And Hermes hosts models on-device via llama.cpp, so prompts need never leave the machine.

> GOVERN
>
> **Let an agent run shell commands unattended without hand-approving each one**
>
> Kiro's local command sandbox contains agent-run shell commands and its autonomous mode auto-approves them inside that boundary, in both CLI and IDE; Codex CLI adds policy-based sandbox integrity checks across bubblewrap, Seatbelt and MXC, so the confinement itself is verified rather than assumed. ADK takes the other path with `require_confirmation` on workflow nodes, pausing for a human only where it matters instead of on every tool call.
>
> Kiro · OpenAI Codex CLI · Google Agent Development Kit

> DEPLOY
>
> **Send the cheap requests to a cheap model without writing the routing logic**
>
> LiteLLM's Auto-Router tiers incoming requests by complexity with pluggable classifiers and reports the resulting cost savings, replacing the hand-rolled if-this-prompt-is-short heuristics most teams bolt onto their gateway. Hermes goes local instead, hosting models on-device through llama.cpp so routine work runs with no provider call at all.
>
> LiteLLM · Hermes

> OPERATE
>
> **Keep a long agent session alive instead of watching it hit the context ceiling**
>
> Claude Code compacts idle conversations automatically, so a session left open across a workday doesn't silently degrade or die at the window limit. Mastra adds conflict-aware Observational Memory writes and run fencing across ten storage adapters, which is what stops two concurrent runs from corrupting the same agent's memory.
>
> Claude Code

BUILD

◆ AI Coding Agents

Anysphere Cursor

Feedback ⓘ snapshot-20261014 NOTES ↗

RANK

Cursor is a built to make you extraordinarily productive, agents turn ideas into code. Accelerate development by handing off tasks to Cursor.

> Cursor added Gemini 4 Argon as a new supported model.

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

01 **Gemini 4 Argon model support** NEW 35

Cursor adds Gemini 4 Argon as an available model.

— Names the model but no usage detail or mechanism product docs

NEW INTEGRATION Origin App Marketplace — security integrations

NEW INTEGRATION Origin App Marketplace — deployment integrations

NEW INTEGRATION Origin App Marketplace — CI integrations

NEW INTEGRATION Origin App Marketplace — code review integrations

NEW INTEGRATION Origin App Marketplace — monitoring and project management integrations

> snapshot-20261014 NOTES ↗ LATEST STABLE
>
> Origin App Marketplace launches with public submission support; apps authenticate via app JWT and installation access tokens against the Origin API.
>
> Launches the Origin App Marketplace, open to public submissions from any developer tooling team.
>
> Apps built against the Origin API authenticate with an app JWT and installation access tokens, receive webhook events, and write back check runs, pull request comments, and reviews.

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.1233 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.

> Magpie's two releases this window add built-in TLS for the gateway and web UI, a new `magpie statusline` command for Claude Code, a daemon flag to serve the web UI, and a speech-to-text passthrough endpoint, alongside several smaller GUI and routing refinements.

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

01 **Built-in TLS for the gateway and web UI** NEW 95

`magpie serve`, `magpie web`, and `magpie daemon install` gain `--tls-cert` and `--tls-key` flags plus `MAGPIE_TLS_CERT` / `MAGPIE_TLS_KEY` environment variables to serve the gateway and web UI over HTTPS. Certificates are re-read automatically when the underlying files change, so a certbot renewal takes effect without restarting the process.

Run magpie as a remote relay over HTTPS — useful when hosting on a server reachable over the internet; certbot renewals are picked up without restarting.

```
$ magpie serve --tls-cert
/etc/letsencrypt/live/example.com/fullchain.pem --tls-key
/etc/letsencrypt/live/example.com/privkey.pem
```

Supply TLS credentials via environment variables instead of flags, e.g. in a systemd unit or container.

```
$ export
MAGPIE_TLS_CERT=/etc/letsencrypt/live/example.com/fullchain.pem
export
MAGPIE_TLS_KEY=/etc/letsencrypt/live/example.com/privkey.pem
magpie serve
```

— Names exact flags, env vars, and reload behavior with runnable commands v0.1.1232

02 **magpie statusline command for Claude Code** UPDATED INTEGRATION

NEW 65

Adds a `magpie statusline` command that prints the session's current model and each account's window counts, intended for embedding in Claude Code's status line.

Embed the current model and account window counts in your shell prompt or tmux statusline for live Claude Code session visibility.

```
$ magpie statusline
```

— Exact command given but behavior detail is brief v0.1.1231

03 **Daemon-served web UI via --web flag** NEW 65

`magpie daemon install --web` serves the browser page directly from the daemon service, giving a persistent link without managing a separate process.

Install the magpie daemon so it also serves the web UI, giving you a stable browser link without a separately managed process.

```
$ magpie daemon install --web
```

— Named flag with runnable install command v0.1.1231

THINNER COVERAGE BELOW

04 **Speech-to-text passthrough endpoint** NEW 55

Adds `/v1/audio/transcriptions` to pass speech-to-text requests through to the underlying provider.

— Endpoint named but no example call or scope detail v0.1.1231

05 **Per-key Tavily usage tracking in GUI** NEW 45

The GUI now shows per-key Tavily usage, displaying what each Tavily key has consumed against its limit, with refused keys automatically rested.

— Describes behavior but not where in the GUI to find it v0.1.1231

06 **400K and 512K context window options** IMPROVED 45

The Context window field gains 400K and 512K options, alongside the existing 128K, 200K, 256K and 1M choices.

— Exact values given but only a UI field, no config key v0.1.1231

07 **Routing group timeout control and write-speed telemetry** NEW 45

Routing groups gain a Slow to start timeout option accepting user-defined minutes alongside the existing Wait, 30 s, 1 min and 2 min presets, and the tray panel's Routing rows now show write-speed telemetry indicating how fast each finished request wrote.

— Two routing-group additions named but only UI location given v0.1.1231

08 **Plugin-declared switches in provider editor** NEW 30

Plugins can now declare switches that appear in the plugin's provider editor and are read by the plugin at runtime.

— Mechanism named only in brief, no concrete example v0.1.1231

09 **Address prompt in refused allow-list dialogs** NEW 25

Refused allow-list dialogs now show an `Add <addr>` prompt naming the address magpie sees the page from.

— Bare description of a dialog change, no further mechanism v0.1.1231

10 **Kimi Code transcript sessions** NEW 25

Enables Kimi Code to open sessions as a transcript.

— One-line mention with no mechanism or scope v0.1.1231

11 **Pace shown as a number on window cards** IMPROVED 15

A window's Pace is now shown as a number on its card and in the menu bar.

— Bare UI change with no further explanation v0.1.1232

NEW INTEGRATION Gemini 4 Argon added to built-in Factory provider

UPDATED INTEGRATION Factory plugin updated to 0.1.26 with Gemini route and byte limit reporting

NEW INTEGRATION Grok Build and Grok App hook integration

UPDATED INTEGRATION Factory model picker updated with Haiku 5.5 and Mistral Large 4

NEW INTEGRATION Nous Portal vendor preset

v0.1.1233 CODE ↗ LATEST STABLE

Adds `magpie quota refresh` and `magpie usage --last` / `--days` CLI commands, N-hour key-limit cycles, ctrl+r TUI model refresh, and search query masking.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.1233 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1233
```

Adds `magpie quota refresh <provider> [<account>] [--json]` CLI command to re-read a provider quota card from the terminal, mirroring the Usage page refresh button.

Adds `--last <dur>` and `--days <n>` flags to `magpie usage`, optionally combined with `--csv` and `--account`, to filter usage to a trailing window ending now.

Adds support for gateway key limits that reset on an N-hour cycle (e.g. every 5 hours) via the access/GUI/CLI key-limit settings.

Adds `ctrl+r` keybind in the TUI provider models view to re-fetch the model list and display the result.

Masks web search queries before they reach a search API for Kimi Code and WorkBuddy.

+5 more

v0.1.1230 CODE ↗ STABLE

Adds Grok hook support via `config.toml`, passthrough video fields for provider APIs, and web-search results in Responses replies.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.1230 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1230
```

Passes unknown video request fields to a provider's own videos API via an `Extra` map, so provider-specific fields like Agnes's `mode` are forwarded as sent.

Preserves the web search that ran — including its query and sources — when translating a Responses reply to a client.

Forwards a gateway plugin's `web_search` field to the client when the plugin's config hook declares `searches: true`.

Allows each account to pick its own recorded allowance cycle, with meters and legend following that selection.

Routes an image request to the next member of a routing group when one member answers 5xx.

+2 more

v0.1.1229 CODE ↗ STABLE

Balance fields now aggregate list values via sum/max/min/count; dry-run quota windows are tracked; hook descriptions render in any language in the GUI.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.1.1229 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.1229
```

Extends `balance` field expressions to aggregate list items — `a.*.used` or `a[status=active].used` — via sum(), max(), min(), or count() (named `balanceExpr` in source); unreadable amounts now render as `—` instead of failing the whole display.

Adds hook description display in any language, shown beneath the hook's name in the GUI.

Tracks dry run windows where a quota was exhausted, flagging calls that may have been paid from credits or extra usage rather than from the subscription window.

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.34.14 NOTES ↗ RANK

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

> Letta Code's only update this window adds a way to name the client when connecting to MCP servers.

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

01 Named client identification for MCP connections NEW

75

The `letta mcp connect` command gains a `--client-name` flag to specify a custom client name when establishing an MCP connection, letting server-side logs and dashboards identify the agent by name.

Connect to an MCP server with a recognizable client name so server-side logs and dashboards identify this agent by name.

```
$ letta mcp connect --client-name my-letta-agent
```

— Names exact flag and command; usage example given, no further mechanism. v0.34.14

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.163.0-alpha.11 NOTES ↗

CODE ↗ RANK

OpenAI Codex CLI runs an agent in the terminal that reads, changes, and tests code in local repositories.

> Codex CLI's two alpha releases add daemon restart with feature-flag overrides, a new installer environment variable for choosing the release source, and policy-based sandbox integrity checks across the bubblewrap, Seatbelt and MXC backends, alongside several smaller TUI and default-enablement changes.

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

01 **Daemon restart with feature overrides** NEW 90

Applies feature overrides when restarting the app-server daemon via `codex app-server daemon restart`, forwarding CLI `--features` flags through to the restarted process.

— Full command and flag documented, directly runnable. rust-v0.163.0-alpha.11

02 **Installer source override and WinGet daemon installation** NEW 85

Adds `CODEX_INSTALLER_USE_RELEASES_OPENAI_COM` environment variable to force the standalone installer to use GitHub Releases instead of `releases.openai.com`, and supports daemon installation from Windows WinGet packages.

Force the Codex installer to pull from GitHub Releases instead of the default OpenAI CDN, useful in environments that block `releases.openai.com`.

```
$ curl -fsSL https://chatgpt.com/codex/install.sh |
CODEX_INSTALLER_USE_RELEASES_OPENAI_COM=false sh
```

— Exact env var name plus a runnable install command. Names the package manager but gives no install command.

rust-v0.163.0-alpha.10 rust-v0.163.0-alpha.11

THINNER COVERAGE BELOW

03 **TUI keyboard selection and shortcuts** IMPROVED 45

Adds keyboard text selection in the transcript view, supports Command/Super keyboard shortcuts in the TUI keymap, and extends copy-on-select to composer text.

— Names UI areas affected but no keys or commands given. rust-v0.163.0-alpha.11

04 **Remote cwd filter in resume picker** IMPROVED 40

Preserves the remote `cwd` filter in the in-app resume picker.

— Names the filter and UI surface but no steps. rust-v0.163.0-alpha.11

05 **CLI Daybreak enabled by default** IMPROVED 30

Enables CLI Daybreak by default and enforces background-turn eligibility.

— No detail on what Daybreak does or its scope. rust-v0.163.0-alpha.11

06 **Cyber access programs for API-key sessions** IMPROVED 25

Enables Cyber access programs for API-key sessions by default.

— Bare statement with no mechanism or scope. rust-v0.163.0-alpha.11

07 **Clickable Markdown links in Rex** IMPROVED 25

Renders Markdown web links as clickable labels in Rex.

— Thin one-line mention with no further detail. rust-v0.163.0-alpha.11

08 **Preference for remote Visualize installs** IMPROVED 25

Prefers remote Visualize installs over the bundled plugin.

— Thin description, no mechanism or configuration named. rust-v0.163.0-alpha.11

WAS THIS USEFUL?  

Anthropic Claude Code

Feedback ⓘ v2.1.299 NOTES ↗

CODE ↗ RANK

Claude Code is Anthropic's terminal coding agent that plans, edits, and tests code in local repositories.

> Claude Code v2.1.299 adds automatic idle compaction for long conversations, finer control over file-mention content and gateway attribution headers, several mod hook and UI component API additions, and new operational flags for plugin eval, self-hosted runners, and agent view.

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

01 Idle compaction for long conversations NEW

91

Conversations exceeding 200K tokens that sit idle for about 54 minutes are now compacted in the terminal before the one-hour prompt cache expires; the threshold is configurable via `CLAUDE_CODE_IDLE_COMPACT_MIN_TOKENS` or disabled entirely with `"idleCompaction": false` in `settings.json` (requires v2.1.290+).

Keep idle long-context sessions cache-warm by letting Claude Code compact them automatically before the prompt cache expires, or raise the threshold to 300K tokens for bigger conversations.

```
$ CLAUDE_CODE_IDLE_COMPACT_MIN_TOKENS=300000 claude
```

Disable idle compaction entirely for a project where you always want full context replayed, via the project settings file.

```json
{"idleCompaction": false}
```

— Full mechanism, thresholds, and config keys named with runnable examples v2.1.299

02 Mod hook API enhancements IMPROVED

87

Mod hooks gain several additions: `$.prompt.media` lets a `prompt.submit` hook read the bytes of an image or document attached to the prompt; the `skill.prompt` hook now also receives the skill's frontmatter, source, plugin name, and file path; a `queue` door on the `session.append` hook lets a mod make the transcript's queue records differ from the prompt as sent; and `/plugin` and `claude plugin validate` output now notes a mod's hooks that are skipped when an organization manages Claude Code.

— Names every hook, field, and command surface changed v2.1.299

03 **Graceful shutdown for self-hosted runner orchestrator** NEW

83

The `claude self-hosted-runner orchestrator` command adds `--shutdown-drain-sec`: on SIGTERM it stops polling and allows the in-flight poll and running spawn hooks up to that many seconds to finish before exiting.

— Named flag with exact SIGTERM behavior described v2.1.299

04 **File-mention content control via environment variable** NEW

78

Setting `CLAUDE_CODE_DISABLE_FILE_MENTIONS` makes Claude receive only the file paths referenced by `@` mentions instead of their full contents, reducing context size and incidental data exposure.

Prevent Claude from ingesting large files inline when a user types `@` mentions — useful for keeping context lean and avoiding accidental data exposure.

```
$ CLAUDE_CODE_DISABLE_FILE_MENTIONS=1 claude
```

— Exact env var and behavior given with runnable example v2.1.299

05 **Tool exclusion in plugin eval runs** NEW

73

The `claude plugin eval` command gains a `--disallow-tools` flag and a `disallowed_tools` case field to exclude specific built-in tools (e.g. Agent) from eval runs.

— Exact flag and field named, clear usage v2.1.299

06 **Session identity lookup for managed-settings mods** NEW

73

Adds $.session.identity() for managed-settings mods to retrieve how the session is signed in, plus organization and email when `forceLoginOrgUUID` lists that organization.

— Named API and returned fields but no usage example v2.1.299

07 **New styling properties for mod Text and Box components**

IMPROVED

73

A mod's `Text` component gains `size`, `weight`, `mono`, and `tabular` properties, and its `Box` component gains named spacing steps (`xs` to `xl`); `bold` is deprecated in favor of `weight`.

— Names every property and deprecation but no code sample v2.1.299

08 /effort and /fast slash commands in agent view NEW

69

New `/effort` and `/fast` slash commands set what new sessions start with; invoking bare `/model`, `/effort`, or `/fast` now opens a picker.

— Exact commands and behavior named, no example given v2.1.299

09 Attribution header suppression via gateway policy NEW

65

Setting `CLAUDE_CODE_ATTRIBUTION_HEADER=0` in a Claude apps gateway `code` policy `env` suppresses the attribution block in Claude Desktop's Code tab and in Claude Code gateway sessions.

— Named env var and scope but no setup walkthrough v2.1.299

THINNER COVERAGE BELOW

10 Safe mode indicator in agents view peek panel NEW

55

Background sessions started with `--safe-mode` now show a safe mode indicator line in the agents view peek panel.

— Names flag and UI location but thin detail v2.1.299

WAS THIS USEFUL? 👍 👎

Token Optimizer

Feedback ⓘ v5.13.43 NOTES ↗ RANK

Token Optimizer is a coding-agent plugin that trims wasted context, checkpoints work across sessions and compactions, and reports token spend on a live dashboard.

> Token Optimizer v5.13.43 adds a trustworthy `--dry-run` mode across install, uninstall, and purge, plus output-filter support for Devin Desktop's Local agent.

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

01 **Dry-run for install, uninstall, purge** IMPROVED 78

`--dry-run` now covers `install`, `uninstall`, and `purge` operations, refusing in exactly the states where the real command would and writing nothing to disk, so it can be used to safely preview destructive operations like a purge.

Safely preview a purge without touching the filesystem — useful before cleaning up a broken or interrupted install.

```
$ token-optimizer purge --dry-run
```

— Names exact subcommands and a runnable example command v5.13.43

THINNER COVERAGE BELOW

02 **Devin Desktop output-filter support** NEW 57

Adds output-filter support for Devin Desktop's Local agent on macOS, compressing a 125 KB output down to ~0.5 KB with full recovery.

— Gives concrete compression numbers but no config/flag to invoke it v5.13.43

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.90 NOTES ↗

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

v0.9.90 adds a `post-merge` rebuild hook via `graphify hook install` and splits hub-fallback labels into a `.graphify_labels.pending.json` sidecar.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.90 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.90
```

Adds `graphify hook install` support for a `post-merge` hook that triggers an unscoped background rebuild after a merge, clearing phantom nodes resurrected by the `graph.json` union merge driver.

Separates hub-fallback community labels into a `.graphify_labels.pending.json` sidecar, keeping `.graphify_labels.json` exclusively for curated (manual/LLM) names; promoted names drop from the sidecar and an emptied sidecar is removed automatically.

UPDATED SUPPORTED PLATFORM C# foreach loop-variable type resolution

└──▷ BREAKING ON UPGRADE

! The AST cache schema is bumped; the first build after upgrading to v0.9.90 will re-extract all sources from scratch.

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-10-06 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 new part kinds for exploded axonometric diagrams, including `vault`, `tools`, `skills`, `harness`, `interface`, `pcb`, `plate`, `keycaps`

Adds an offline font option so generated output works without loading Google Fonts.

Adds a WCAG contrast gate in CI to enforce accessible colour contrast.

WAS THIS USEFUL? 👍 👎

Yuval Sarel cones

Feedback ⓘ v0.3.15 NOTES ↗ RANK

A terminal workspace for coding agents: peek into Claude Code, Codex, pi and OpenCode sessions, return to past work, and start work in another project without leaving the dashboard.

> cones shipped multi-home session discovery, a new `cones stop` command for terminating headless runs, a `cones launch` command for detached Codex sessions, a `plan` column surfacing billing/provider info, new `cones config` inspection and mutation subcommands, and a bundled `cones` skill for agent self-configuration, alongside dozens of smaller dashboard, Composer, and session-list refinements across 14 releases.

└──▷ **WHAT SHIPPED · 36 FEATURES** most completely described first ? what's the number?

01 **Multi-home session discovery** NEW 93

Adds a `homes` block in `jobs.yaml` to register additional home directories per harness, configurable via `cones config set homes.codex '[/path/to/home]'`, with wildcard suffix matching on `homes` entries (e.g. `/tmp/runs/codex-home.*`) so live clients and finished-run history are discovered across dynamically named directories. Sessions started with their own `CLAUDE_CONFIG_DIR`, `CODEX_HOME`, or similar environment variables are now listed and read in place, with live clients under the current `HOME` picked up automatically.

— Exact config command and env vars given; fully actionable. v0.3.14 v0.3.12

02 **`plan` column for provider and billing info** NEW 93

Adds `plan` column to the dashboard (off by default) showing the serving/billing provider per session — Claude Code reports `bedrock` or `vertex` for cloud deployments, otherwise the signed-in account plan (`pro`, `max`, `team`, `enterprise`), or `api` with no account; Codex shows the ChatGPT plan from its rate-limit headers or falls back to its model provider; pi and OpenCode show their named provider. Enable via the column picker or by adding it under `columns:` in `jobs.yaml`.

— Exact column name, values, config key, and enable path. v0.3.11

03 **cones skill for agent self-configuration** IMPROVED 90

Installs a `cones` skill for Claude Code, Codex, pi, and OpenCode via `~/.agents/skills` when the dashboard starts, exposing capabilities including session visibility, inter-session messaging, launching work, folder handoff to a coordinator, settings changes, and problem reporting (inspect with `cones skill cones`). The skill later gained a bundled `cones skill cones` variant letting an agent configure homes automatically, and was expanded to report a folder's roster, how replies are returned (`comms wait`, `comms mail`, `mail --ack N`), and that `cones ls --json` prints one object per line.

— Names install path, inspect command, and every reply mechanism. v0.3.12 v0.3.8 v0.3.7

04 **Id-prefix resolution for comms commands** NEW 86

Adds unambiguous id-prefix support (four or more characters) to `cones comms send` and `cones comms wait --id`, mirroring how `cones show` already works — an unmatched prefix names the problem and points at `cones ls --dir`; an ambiguous prefix lists its matches.

— Exact commands, flag, and fallback command named. v0.3.8

05 **cones config inspection and mutation commands** NEW 86

Adds `cones config` command to print the `jobs.yaml` path and its reference, with `--check` to validate the file, plus `cones config set` and `cones config unset` subcommands to change one setting by dotted path, e.g. `cones config set worktrees flat`.

— Exact subcommands, flag, and example invocation given. v0.3.1

06 **Shell alias environment variable routing** IMPROVED 83

Reads `NAME=value` prefix variables from your shell alias (e.g. `AWS_PROFILE` and `CLAUDE_CODE_USE_BEDROCK=1`) when launching a session, so provider routing follows the alias instead of the native login. Documents that sessions reopened via Claude's background daemon inherit the daemon's own environment, and recommends placing provider variables in the `env` block of `~/.claude/settings.json` for full coverage.

— Names exact env vars and config file path for full coverage. v0.3.4

07 **`cones launch` command for detached Codex runs** NEW 81

Adds `cones launch` command to start Codex detached and return its thread id, with `--no-prompt-record` flag to keep the prompt out of `launches.jsonl`.

— Exact subcommand and flag, runnable command. v0.3.12

08 **`cones stop` for headless runs** NEW 80

`cones stop` terminates a headless run — such as `codex exec` or `claude -p` under a dispatcher — by checking the process, sending `SIGTERM`, and waiting for exit. It now also matches a client whose environment cannot be read to the home that holds its conversation, replacing the bare process row fallback, extending `cones stop` to sessions in remote homes.

— Named command and mechanism, extension across releases described. v0.3.14 v0.3.13

09 **Dashboard row listing and labeling** IMPROVED 78

`cones ls` now prints column names (id, job/folder, status, started, harness, cost, reason/tokens) to stderr, keeping stdout rows-only. `codex exec` runs are listed regardless of the first word of the prompt, including prompts starting with `update`, `apply`, `debug`, or `help`, and the npm-installed Codex node wrapper no longer appears as its own row. A Codex row with no reported model now displays the `-m` / `--model` flag it was started with, falling back to the rollout value once written, and a `codex exec` row with no rollout is titled by its prompt, omitting the subcommand and its options.

— Enumerates every column and row-labeling rule changed. v0.3.15 v0.3.14

---

10 **Startup config keys for prompt length and update check** NEW

78

Adds `start.min_prompt` config key to refuse composer instructions shorter than a configured minimum length, and `start.update_check` config key to suppress the new Homebrew release announcement shown in Help.

— Both config keys named with exact behavior. v0.3.0

---

11 **`defaults.skip_permissions` global config key** NEW

75

Adds `defaults.skip_permissions` to the global config, letting you turn permission prompts on or off for every harness the composer starts; per-harness keys like `claude_skip_permissions` still override it.

— Exact global and per-harness config keys named. v0.3.2

---

12 **Shell-like completion for add-folder input** IMPROVED

70

`+ add folder` input now completes like a shell: Tab expands the shared prefix, lists matches, and appends `/` to `~` or `..`; Enter on a missing path prompts, and a second Enter creates it with its parents.

— Exact keys and behavior for a UI input, no command name. v0.3.9

---

13 **Worktree flat/indented session grouping** NEW 70

Adds `worktrees: flat` config key in `jobs.yaml` to control whether linked worktrees are listed flat or under an indented `⑂` heading inside their repository folder; by default, linked worktrees with two or more sessions get this indented `⑂` heading in the session list.

— Names config key and default heading behavior. v0.3.1

---

14 **`children` and `tasks` dashboard columns** NEW 65

Adds `children` column (also accepted as `commands`) to the dashboard, on by default, showing processes running under a session, and adds `tasks` column, on by default,

showing what Claude reports for a background job by kind (e.g. `2 shells 1 monitor`), matching its footer.

— Names both columns and example output, no config key given. v0.3.9

15 **Homebrew bottle for Apple silicon** IMPROVED 65

Adds a prebuilt Homebrew bottle for Apple silicon via `brew install cones`, eliminating the need for Rust or a current Xcode on Mac.

— Exact install command, clear before/after. v0.3.6

16 **Cross-harness session forking** NEW 65

Cross-harness session forking via `Ctrl+Y` lets users seed a different harness with an existing conversation; forks list under their parent with a fork arrow.

— Exact keybinding and resulting UI behavior. v0.3.0

THINNER COVERAGE BELOW

17 **Composer draft cursor and navigation** NEW 55

Adds click-to-position cursor support in the Composer draft, moving the cursor to the character under the click, and adds multi-row draft navigation: Up and Down move between rows, reaching the session list only from the first and last row.

— Describes behavior but no config surface to invoke. v0.3.13

18 **`--print-command` requires a running Codex daemon** BREAKING 55

`--print-command` no longer starts the Codex daemon; with none running it now fails and names the command that starts one.

— Exact flag and new failure behavior named. v0.3.12

19 **Fullscreen composer exit keys** IMPROVED 55

Left and Tab now return from Claude Code's fullscreen composer when `CLAUDE_CODE_NO_FLICKER=1` is set, instead of reaching Claude and backgrounding the session.

— Names exact env var and keys, narrow behavior scope. v0.3.8

20 **Image drag-and-drop in composer** NEW 55

Supports dragging image files from Finder into the dashboard composer, or pasting their paths, converting them to `[Image #n]` markers identical to pasted clipboard images.

— Describes mechanism and marker format, no command. v0.3.7

21 **Hosted terminal and cross-harness dashboard visibility** NEW

50

Hosted terminals now show their screen, and Copilot and Cursor conversations can be read from the dashboard; notes also reach pi and OpenCode through their reporters.

— Names the surfaces covered but no config/command given. v0.3.12

22 **Agent-list title recognition for Claude Code 2.1.291** IMPROVED 45

Recognises Claude Code 2.1.291's count-prefixed agent-list titles (e.g. '3 awaiting input · claude agents') so the dashboard correctly opens and closes that list.

— Names the exact version and title format, narrow compatibility fix. v0.3.3

23 **Background run input-status detection** IMPROVED 45

A background run that ends its turn on a question now ends as `input` status and waits for an answer, instead of remaining `started` until timeout.

— Names status value and prior/after behavior. v0.3.1

24 **Stale run reaping** IMPROVED 45

Stale runs whose supervisor died are reaped when the ledger is read, so the dashboard and `cones ls` no longer show them as running.

— Names command affected and mechanism trigger. v0.3.0

25 **Process-tree CPU and memory accounting** IMPROVED 40

CPU and memory metrics now sum a session's entire process tree, stopping at any other listed session below it, giving accurate cost for multi-core builds.

— Explains mechanism but no named surface to act on. v0.3.9

26 **Dashboard divider layout adjustments** IMPROVED 40

Adds a blank column separating the divider from the session list and from the viewer in the dashboard, and supports resizing the divider between the session list and detail pane by dragging.

— UI layout change, no config key, clear navigation only. v0.3.8 v0.3.0

27 **Bypass column for permission-skip sessions** NEW 40

Adds an optional bypass column to the session list marking sessions running without permission prompts.

— Names the column but not how to enable it. v0.3.0

28 **First-run config directory creation** IMPROVED 40

Creates `~/.cones` on first config write for fresh installs instead of failing.

— Names the path but bugfix-level thinness. v0.3.3

29 **Claude-to-Claude messaging routed via SendMessage** IMPROVED 35

A Claude session sending to another Claude session is now routed to `SendMessage` instead of the inbox file.

— Names mechanism change but no operational surface for reader. v0.3.9

30 **Worktree folder session heading on add** IMPROVED 35

A folder opened with `+ add folder` inside a worktree now heads its own sessions, and Enter selects the first one.

— Thin UI behavior, no named config. v0.3.1

31 **Rest-to-open finished background session** IMPROVED 35

Resting on a finished Claude background session now opens it in the pane without requiring Enter.

— Thin single UI behavior change. v0.3.1

32 **Awake-time run duration accounting** IMPROVED 35

A run's duration and timeout now count only awake time, excluding time while the lid is closed.

— Explains mechanism but no actionable surface. v0.3.1

33 **Resumed session folder grouping** IMPROVED 30

Keeps sessions resumed from history grouped under their repository folder rather than their worktree heading.

— Thin behavior statement, no named surface. v0.3.3

34 **Inline Claude trust question** NEW 30

Displays Claude's trust question inline in the pane when a folder is untrusted.

— Single thin UI statement. v0.3.0

35 **Harness switch persistence** IMPROVED 30

Harness switches are persisted to config defaults automatically, with retry on failed save.

— Thin statement, no config key named. v0.3.0

36 **Symlinked folder session pinning** IMPROVED 25

Sessions now list under their folder when the folder is pinned through a symlink.

— Very thin, no mechanism detail. v0.3.0

└──▷ **BREAKING ON UPGRADE**

! `--print-command` no longer starts the Codex daemon; with none running it fails and names the command that starts one.

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.97 NOTES ↗ RANK

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

> Copilot CLI v1.0.97 adds secret-redacting JSON output, agent-scoped declarative hooks for custom agents, and billed AI Credit tracking in `/usage`, alongside improvements to inline image rendering, managed permission rules in `/config`, plugin and custom-agent discovery reliability, and new MCP server support for Power BI and Bluebird.

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

01 Plugin and custom-agent discovery reliability IMPROVED

75

Copilot CLI now discovers all custom agents even when repositories define more than 100 or nest agent files deeply, and ensures skills and custom agents from `--plugin-dir` plugins reach autocomplete and invocation. Remote plugin marketplace and plugin fetches now retry once with your signed-in GitHub token after an HTTPS auth failure.

Load custom agents and skills from a local plugin directory so they appear in autocomplete and can be invoked by name.

```
$ copilot --plugin-dir ./my-agents
```

— Names flag, limit, and retry behaviour with example v1.0.97

02 Secret redaction in JSON task output NEW

65

`--output-format json` now redacts secrets from output while preserving valid variable references, so piped or logged task output does not leak secrets from your environment.

Pipe Copilot task output to a log file or downstream tool without leaking secrets from your environment into the JSON.

```
$ copilot --output-format json
```

— Names exact flag and mechanism, with a runnable example v1.0.97

03 Inline images rendered by default IMPROVED

60

Inline images now render by default on supported terminals; this can be disabled with `/settings inlineImages false`.

Turn off inline image rendering if your terminal displays garbled escape sequences instead of images.

```
$ /settings inlineImages false
```

— Exact toggle command given and demonstrated with an example v1.0.97

THINNER COVERAGE BELOW

04 **Billed AI Credits shown in /usage** IMPROVED 50

`/usage` now shows billed AI Credits for token-billed sessions, including after a session is resumed.

— Names command and behaviour but no further mechanism v1.0.97

05 **Managed permission rules locked in /config** BREAKING 50

Managed permission rules in `/config` are now read-only and hidden unless they have been set by your organization.

— Names exact UI surface and new read-only behaviour v1.0.97

06 **Richer synchronous /task JSON output** IMPROVED 50

Synchronous `/task` JSON output now includes failed child error details and lifecycle events before completion is reported.

— Names command and new fields but no schema shown v1.0.97

07 **MCP server support for Power BI and Bluebird** NEW 45

Adds support for Power BI and Bluebird MCP servers, accepting their published resource scopes during sign-in.

— Names both integrations but no setup steps given v1.0.97

08 **Streaming fix behind TLS-intercepting proxies** IMPROVED 35

Streamed responses behind TLS-intercepting proxies now complete correctly when the `close_notify` signal is omitted.

— Names the specific TLS condition fixed, no further detail v1.0.97

09 **Agent-scoped declarative hooks** NEW 25

Adds agent-scoped declarative hooks for custom agents.

— Bare one-line mention with no mechanism or usage shown v1.0.97

WAS THIS USEFUL?  

claude-mem

Feedback ⓘ v13.38.0 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's v13.38.0 release rewrites Claude Code hooks to exec form to kill Windows console flicker, ships a compiled native launcher for install/uninstall, and adds three new configuration environment variables.

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

01 Native launcher for install, update and uninstall NEW

90

`npx claude-mem install` / `update` now compile and place a native launcher at `~/.local/bin/claude-mem` (macOS/Linux) or `%LOCALAPPDATA%\claude-mem\bin\claude-mem.exe` (Windows), adding that folder to PATH automatically, and `npx claude-mem uninstall` removes the compiled launcher. The launcher fails open: on any error it prints `claude-mem: <reason>, continuing without memory` and lets the session continue.

— Exact install paths and commands given v13.38.0

02 New configuration environment variables NEW

80

`CLAUDE_MEM_DISABLE_SUMMARY=1` skips end-of-session summaries for headless or read-only runs; `CLAUDE_MEM_LLM_STREAM_IDLE_TIMEOUT_MS` configures the streamed-request idle window for self-hosted backends with long queue waits (default: 90 seconds); and `CLAUDE_MEM_OBSERVER_THINKING` controls thinking-parameter behaviour in the observer.

— Names exact env vars with defaults and purposes v13.38.0

03 Exec-form Claude Code hooks BREAKING

77

Every Claude Code runtime hook is rewritten to exec form (`"command": "claude-mem", "args": [...]`) with no shell in between, eliminating the console window flash on Windows. This requires Claude Code 2.1.139 or newer; `install` and `doctor` warn on older versions, though hooks will not function correctly below that version. If the launcher folder was not already on PATH, the first post-upgrade session instructs the user to restart Claude Code from a new terminal before hooks work.

— Names exact version requirement and exec-form mechanism v13.38.0

└──▷ **BREAKING ON UPGRADE**

! Hooks in exec form require Claude Code 2.1.139 or newer; `install` and `doctor` warn on older versions but the hooks will not function correctly below that version.

! After upgrading, if the launcher folder was not already on PATH, the first session will instruct you to restart Claude Code from a new terminal before hooks work.

WAS THIS USEFUL? 👍 👎

JuliusBrussee Caveman

Feedback ⓘ bin-v4.0.1 NOTES ↗

RANK

why use many token when few token do trick. Viral skill + proxy for coding agents that cuts 65% of tokens by talking like a caveman.

> Caveman's headline release this window is 3.0.0, which stabilizes its middleware at 1.0.0 across Python and TypeScript, relicenses the entire toolchain to Apache-2.0, and adds Autopilot re-checking, a Memory Doctor and telemetry opt-out. That was followed by 3.1.0's three-skill restructure, 3.2.0's new language coverage and caveman-compress providers, and first-stable SDK, middleware and contracts packages adding circuit breakers, OTel tracing, protocol 1.1 schemas and version gating.

└──▷ **WHAT SHIPPED · 38 FEATURES** most completely described first ? what's the number?

01 Middleware Protocol 1.1 contracts and OpenAPI spec NEW

93

Adds `middleware-error`, `middleware-session-delete`, `middleware-session-delete-response`, `middleware-receipt-response`, and `middleware-decision-event` message schemas; adds `openapi/middleware.openapi.json` (OpenAPI 3.1) covering `capabilities`, `optimize`, `retrieve`, `receipts`, and `sessions/delete` endpoints; extends `middleware-capabilities` with optional `protocol {min,max}`, `features`, and `max_retention_seconds` fields plus new `limits` keys `retrieve_deadline_ms`, `queue_depth`, `retrieve_queue_depth`, `max_segments`, `max_manifest_items`, `receipt_bytes`, and `quota_requests_per_minute`; adds `feature`, `reason`, and `positive_limit` definitions to `middleware-common`; pins every schema `$id` to an immutable GitHub release tag with relative cross-schema `$ref` paths for fully local resolution; extends `scripts/validate-schemas.mjs` to check `$id` against the file name, validate protocol 1.1 fixture examples, and verify every OpenAPI `$ref`; and closes response schemas (`additionalProperties: false`) so the conformance kit can check servers against exactly what a 1.1 server emits.

— Names every schema, endpoint and limit key `contracts-v2.0.0`

02 Three-skill restructure replacing intensity levels IMPROVED

83

v3.1.0 replaces six intensity levels with three self-contained skills `/caveman`, `/ultracave`, and `/megacave`, adding Simplified Technical English (ASD-STE100) as the baseline register shared across all three. Mode IDs are now `caveman`, `ultracave`, `megacave` (plus one-shots and `off`); the `SessionStart` hook injects the active skill in full and per-turn reinforcement derives from the skill's thesis line.

Adds `defaultMode: "manual"` as a tracked maintenance configuration state alongside `/caveman status`.

– Names config keys and hook mechanism but no example `v3.1.0`

---

03 **Startup compatibility checks and version gating in middleware-python**

NEW

80

Adds preflight() and ready() startup-check methods, with `COMPATIBILITY` listing framework tiers and tested version ranges. Adds an `accept_framework_version=True` override and `version_unverified` / `unsupported_version` reason codes to the version-gate system, plus an `Adapter.version` property exposing the installed package version and `caveman_middleware.__version__` read from the installed distribution.

– Names methods and properties, lacks example call `middleware-python-v1.0.0`

---

04 **LangChain model subpath for middleware-ts** NEW

78

Adds a new `@caveman-ai/middleware/langchain-model` subpath exporting `withCavemanModel`, `CavemanChatModel`, `scopeFromConfig`, and `CavemanDocumentCompressor`, requiring only `@langchain/core` instead of the full `/langchain` bundle.

– Names exact exports and dependency, no import example `middleware-ts-v1.0.0`

---

05 **Contracts package renamed and re-hosted schema IDs**

BREAKING

78

The npm package is renamed from `@caveman/contracts` to `@caveman-ai/contracts` (any `package.json` dependency on the old name must be updated), and every schema `$id` changes from `https://caveman.so/schemas/…` / `https://caveman.cloud/schemas/…` to `https://raw.githubusercontent.com/JuliusBrussee/caveman/contracts-v2.0.0/packages/shared/contracts/schemas/<file>`; tools or validators relying on the old `$id` URLs will no longer match.

– Exact old and new names/URLs for migration `contracts-v2.0.0`

---

06 **Circuit breaker in SDK python** NEW

78

Adds CircuitBreaker.release() and a new circuit breaker that opens after 5 consecutive or 10 of 20 failures, stays open 30 s, then allows one probe; Retry-After is honored and deadlines count as failures.

– Gives exact thresholds and timing, no code sample `sdk-python-v1.2.0`

---

07 **OTel tracing and metrics in SDK python** IMPROVED

78

Adds opt-in OTel `tracer` and `meter` support so requests carry `traceparent` / `tracestate` for the SDK's own span when a `tracer` is configured, adds OTelExporter.record_span(cache_creation_tokens=) emitting `gen_ai.usage.cache_creation.input_tokens`, and emits cost as `caveman.usage.cost_usd` from the exporter; `gen_ai.usage.cost_usd` is deprecated (stays through 1.x) in favor of `caveman.usage.cost_usd`.

— Names fields and deprecation path, no wiring example `sdk-python-v1.2.0`

---

08 **sdk-ts breaking API changes** BREAKING 78

recovery() now returns `null` for an invalid scope instead of its previous behavior; deleteSession() now returns a `SessionDeleteResult` instead of its previous return type; `validateCapabilities` is removed; scopes are normalized so values that are not valid tokens are hashed to `h-…`; and Node.js versions below `22.12` are no longer supported.

— Names exact functions and version floor for migration `sdk-ts-v1.2.0`

---

09 **Expanded provider and framework support in middleware-python** NEW 76

Supports OpenAI 2.x (`httpx`) and 3.x (`httpx2`) in the same adapter, eliminating `No module named 'httpx2'` import failures; supports Anthropic Bedrock and Vertex clients, preserving `aws_profile` on `AnthropicBedrock` so calls are signed with the caller's AWS identity; adds LiteLLM sync Router support for OpenAI and Anthropic providers with `unsupported_provider` for other providers; and adds unwrap_google() to return the original Google client after wrapping.

— Names providers and error strings, no code sample `middleware-python-v1.0.0`

---

10 **SDK ts middleware stabilization with protocol 1.1** IMPROVED 76

`@caveman-ai/sdk/middleware` is promoted to stable (drops the `@experimental` marker) and now follows semver, implementing middleware protocol 1.1 with tolerant capability parsing, single-flight refresh, `Retry-After` support, a circuit breaker, and per-candidate concurrency budgets; the transport now honors `HTTPS_PROXY` / `HTTP_PROXY` / `NO_PROXY` without requiring `NODE_USE_ENV_PROXY`, never proxies loopback, and adds a `ca` option.

— Names mechanisms and env vars, no sample config `sdk-ts-v1.2.0`

---

11 **Middleware reaches stable 1.0.0 (Python and TS)** NEW SUPPORTED PLATFORM NEW 75

`caveman-middleware` and `@caveman-ai/middleware` are stabilized at 1.0.0, requiring `@caveman-ai/sdk` / `caveman-sdk` >= 1.2.0; the Python middleware's first stable release ships multi-framework LLM adapters, version gating, fail-open mode, and

SBOM provenance, while the TS middleware certifies `ai-sdk`, `langchain`, `openai`, and `anthropic` adapters as stable tiers (all others experimental). Outside strict mode, adapter errors send the original request and record an `adapter_error` reason code; strict mode raises `MiddlewareError`.

— Names requirement and adapter tiers, no setup command `middleware-python-v1.0.0` `middleware-ts-v1.0.0` `v3.0.0`

---

12 **Transport layer proxy and concurrency controls in SDK python** NEW 75

Adds `HTTPS_PROXY` / `HTTP_PROXY` / `NO_PROXY` environment variable support, `ssl_context`, a pluggable `transport`, `allow_insecure_transport`, `max_concurrency`, endpoint path prefixes, and keep-alive reuse to the transport layer.

— Names every option, no example configuration `sdk-python-v1.2.0`

---

13 **Configurable version gating and size limits in middleware-ts** IMPROVED 74

Adds an `acceptFrameworkVersion` override to control handling of unsupported or out-of-range framework versions, which now pass content through with an `unsupported_version` warning instead of silently doing nothing, plus configurable `manifestBytes` and `wireBytes` options to control history and image size limits, preventing large histories or earlier images from skipping calls.

— Names options and behaviour change, no defaults given `middleware-ts-v1.0.0`

---

14 **Async sink options for SDK ts constructor** NEW 74

Adds new options `maxConcurrency`, `allowInsecureTransport`, `onDecision`, `tracer`, and `meter` to the SDK constructor, with `onDecision` and `onDiagnostic` support for async sinks where rejected promises are swallowed rather than causing unhandled rejections, plus a new read-only `MiddlewareRuntime.strict` property.

— Names constructor options and property, no snippet `sdk-ts-v1.2.0`

---

15 **CLI telemetry opt-out** NEW 73

Adds `caveman telemetry off` and `DO_NOT_TRACK=1` to opt out of CLI usage stats (on by default; CI never sends).

— Exact command and env var to run today `v3.0.0`

---

16 **Memory Doctor diagnostic** NEW 73

Introduces a Memory Doctor that detects `MEMORY.md` notes Claude Code cannot load (beyond the first 200 lines or 25 KB), broken `@imports`, duplicate rules across files, and stale paths in `CLAUDE.md`.

— Names exact files and limits, no invocation command `v3.0.0`

---

17 **Autopilot background re-check for Learn** NEW 72

Adds `caveman learn autopilot off` to control background re-checking of session history, which runs at most every 6 hours at low priority without blocking sessions.

– Exact command and schedule given v3.0.0

---

18 **Standalone providers and report compression in caveman-compress**

NEW 71

Adds standalone `OpenCode` and OpenAI-compatible providers to `caveman-compress`, with provider-specific model selection and protected `nocompress` regions, and supports compression of `JUnit`, `pytest`, and `Jest` test reports.

– Names providers and formats but no exact command v3.2.0

---

19 **caveman_retrieve refusal handling** IMPROVED 71

A `caveman_retrieve` call refused by the runtime (unknown/expired handle, runtime down, or invalid arguments) now returns `{"error":"<code>"}` or an MCP `isError` result instead of crashing the native tool loop.

– Names exact return shape and causes, no sample payload middleware-ts-v1.0.0

---

20 **Middleware-ts reliability fixes** IMPROVED 71

The `fetch` option on OpenAI and Anthropic wrappers is now optional and defaults to the client's own `fetch`; wrapping a client, agent, or model twice with `withCaveman` is now idempotent, running one Caveman layer instead of disabling compression or throwing; entry points that can only record now report `recovery_unbound` at construction; and in strict mode, adapter exceptions now raise MiddlewareError('adapter_error') instead of passing through silently.

– Lists four named fixes, no reproduction steps middleware-ts-v1.0.0

---

21 **SDK python ergonomics, reason codes and compatibility**

IMPROVED 71

Adds as_sync(), ensure_sync(), and ensure_async() helpers; warn-once logging on `caveman.middleware` and `on_decision` events (including `no_candidate`); new `unsupported_provider` and `unsupported_request` reason codes; decline() now accepts any catalog reason and an optional adapter id and never raises in strict mode, with strict ready()/preflight() reporting the first decline(); `MiddlewareError` takes an optional detail argument; and the Python floor is lowered from 3.13 to 3.11.

– Names every helper and code, no single example sdk-python-v1.2.0

---

22 **Pi ChatGPT OAuth proxy routing** NEW INTEGRATION NEW

70

Routes Pi's ChatGPT OAuth traffic through the subscription proxy with `zstd` request decoding, passes through rate-limit responses with `Retry-After` while pinning accepted raw retries to their conversation lineage, and records cache-prefix caveman-caused changes in local request telemetry and `caveman status`.

— Describes mechanism and named command, no step-by-step `v3.2.0`

---

23 **Warn-once log line format changed** BREAKING 70

The warn-once log line prefix changed from `Caveman middleware decision:` to `Caveman middleware passed content through unchanged: adapter=… reason=…`; log filters matching the old prefix must be updated.

— Exact old and new log strings for migration `sdk-python-v1.2.0`

---

24 **Manifest byte budget for long histories** NEW 65

Adds a `manifest_bytes` budget (2 MiB default) for long histories, replacing skipped calls with hashed bytes and images.

— Names config key and default, no usage example `middleware-python-v1.0.0`

---

25 **caveman_cloud.middleware implements protocol 1.1** NEW 65

`caveman_cloud.middleware` implements middleware protocol 1.1 and is now stable, following semver; requests carry `Caveman-Middleware-Features` and `Caveman-Middleware-Client` headers.

— Names headers, no endpoint or sample request `sdk-python-v1.2.0`

---

26 **Apache-2.0 relicense across the toolchain** BREAKING 60

Relicenses the entire repo — engine, proxy, browse, MCP server, shrink, cavemem Go core, and shared platform — under Apache-2.0 with no hosted-service restriction and no commercial license requirement (v3.0.0); this applies to the `middleware-python`, `middleware-ts`, `sdk-python`, `sdk-ts`, and `contracts` packages as of their respective 1.x/2.x releases, with releases before these keeping the MIT license.

— Lists affected components, no migration steps `middleware-python-v1.0.0`

`middleware-ts-v1.0.0` `sdk-python-v1.2.0` `contracts-v2.0.0` `sdk-ts-v1.2.0` `v3.0.0`

---

27 **TypeScript CJS resolution shims** NEW 60

Adds `typesVersions` and `.d.cts` shims so TypeScript projects using `node10` or `node16` CommonJS resolution can import both entry points without `skipLibCheck`.

— Names exact resolution modes, no import path `sdk-ts-v1.2.0`

THINNER COVERAGE BELOW

28 **Standalone installer relocated to installer/** BREAKING

55

The standalone installer has moved from `bin/` to `installer/` for enterprise marketplace compatibility; any scripts or workflows referencing the old path must be updated.

— Exact old and new paths given for migration v3.2.0

29 **Postgres HA store and TLS/mTLS for the proxy** NEW

55

Adds a Postgres HA store, TLS/mTLS support, and HA deploy manifests under `deploy/` to the proxy.

— Names deploy path, lacks operational detail v3.0.0

30 **Voice skills for Claude and Codex installs** NEW

50

Native Claude and Codex installs include three voice skills with subagents inheriting their session's mode, while headless `claude -p` and Agent SDK sessions start with voice off.

— Describes behaviour and defaults, no config key v3.2.0

31 **Always-on integration coverage for more coding agents** NEW

47

Adds or extends always-on integration for Codex, Cursor, Copilot CLI, Antigravity CLI, Grok Build, and `CodeBuddy Code` (v3.2.0), including Grok Build install support added via contributor PR #1084 (v3.1.0).

— Names integrations but no setup steps v3.2.0 v3.1.0

32 **Session stats recording and trend reporting** NEW

47

Records session stats at `SessionEnd` with project-scoped session lookup (v3.2.0), and adds week-over-week session trend reporting using median session data, shown when at least 5 sessions exist (v3.0.0).

— Explains trigger and threshold, no UI path v3.2.0 v3.0.0

33 **Language and format coverage expansion** NEW

45

Adds `tree-sitter` language support to the engine (v3.2.0), plus GDScript elision support (contributor PR #1076), a C++ parse gate (contributor PR #1078), and syslog severity handling (contributor PR #1156) (v3.1.0).

— Lists language additions without usage detail v3.2.0 v3.1.0

34 **Per-request scoping with identifier normalization** NEW

45

Every adapter accepts a per-request scope function, with emails and free-text ids normalized at the adapter level.

— States mechanism briefly, no code example middleware-ts-v1.0.0

35 **Deterministic fidelity evaluation suite** NEW 38

Adds a deterministic fidelity suite and language-specific prompt sets for evaluation, retaining partial results after interrupted runs.

— Describes purpose, no command or path given v3.2.0

36 **OpenCode subagent installation** NEW 35

Installs OpenCode cavecrew agents as subagents and warns when an active provider bypasses the proxy.

— Brief description, no configuration detail v3.2.0

37 **Bundled model price list** IMPROVED 33

Bundles a built-in model price list covering current Claude and GPT lineups, no longer dependent on the working directory.

— States change, no file path or command v3.0.0

38 **Browser extension 1.3.0 editor improvements** IMPROVED 25

Browser extension 1.3.0 improves rich-editor snapshots and send-control discovery.

— Thin one-line description with no detail v3.2.0

NEW SUPPORTED PLATFORM Signed runtime companion binaries for macOS, Linux, and Windows

NEW INTEGRATION Sonnet 5.5, Haiku 5.5, GPT-6.1 Sol, GPT-6 Luna model catalog entries

└──▷ **BREAKING ON UPGRADE**

! The standalone installer has moved from `bin/` to `installer/`; any scripts or workflows referencing the old path must be updated.

! Relicensed from MIT to Apache-2.0 as part of Caveman 3.0.0; releases before middleware-python-v1.0.0 keep the MIT license.

! Relicensed from MIT to Apache-2.0; releases before middleware-ts-v1.0.0 retain the MIT license.

! In strict mode, adapter exceptions now raise MiddlewareError('adapter_error') instead of passing through — callers that previously caught pass-through behavior in strict mode will see a thrown error instead.

! Relicensed from MIT to Apache-2.0; releases before sdk-python-v1.2.0 keep the MIT license.

- ! The warn-once line prefix changed from `Caveman middleware decision:` to `Caveman middleware passed content through unchanged: adapter=… reason=…`; log filters matching the old prefix must be updated.
- ! `gen_ai.usage.cost_usd` is deprecated (stays through 1.x); use `caveman.usage.cost_usd` instead.
- ! The package is relicensed from MIT to Apache-2.0; releases before this one keep the MIT license.
- ! The npm package is renamed from `@caveman/contracts` to `@caveman-ai/contracts`; any `package.json` dependency referencing the old name must be updated.
- ! Every schema `$id` changes from `https://caveman.so/schemas/…` / `https://caveman.cloud/schemas/…` to `https://raw.githubusercontent.com/JuliusBrussee/caveman/contracts-v2.0.0/packages/shared/contracts/schemas/<file>`; tools or validators that relied on the old `$id` URLs will no longer match.
- ! Relicensed from MIT to Apache-2.0; releases before sdk-ts-v1.2.0 retain the MIT license.
- ! recovery() now returns `null` for an invalid scope instead of its previous behavior.
- ! deleteSession() now returns a `SessionDeleteResult` instead of its previous return type.
- ! `validateCapabilities` is removed.
- ! Scopes are normalized: values that are not valid tokens are hashed to `h-…`.
- ! Node.js versions below `22.12` are no longer supported.
- ! The SDK license changes from MIT to Apache-2.0 as of this release.
- ! `caveman-middleware` 1.0.0 requires `@caveman-ai/sdk` / `caveman-sdk` >= 1.2.0; earlier SDK versions are no longer supported.

`bin-v4.0.1` NOTES ↗ **LATEST** STABLE

Caveman v4.0.1 ships signed binaries for macOS, Linux, and Windows; the ghcr.io container image stays Go-only while the standalone binaries gain multi-language compression.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch bin-v4.0.1
https://github.com/JuliusBrussee/caveman.git
# already have the repo? check out this version:
$ git checkout bin-v4.0.1
```

Extends `caveman-proxy` and `caveman-engine` binary code compression to support TypeScript, JavaScript, Python, Rust, Java, and C/C++ (the `ghcr.io caveman-proxy` container image remains Go-only).

Releases signed runtime companion binaries for macOS, Linux, and Windows under the Apache License 2.0.

v4.0.0 NOTES ↗ STABLE

Caveman 4.0.0 unifies install under `npx @caveman-ai/cli`, adds Auto cloud routing via `caveman-auto` with a 1M context window, new CLI commands, prompt-cache warming, and Caveman Blocks as a module.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v4.0.0
https://github.com/JuliusBrussee/caveman.git
# already have the repo? check out this version:
$ git checkout v4.0.0
```

Adds `npx @caveman-ai/cli` as the single install entry point, with a one-screen setup that auto-detects coding agents and Claude Code profiles, applying config on Enter or non-interactively with `--yes`.

Adds `caveman stop`, `caveman doctor` (offline), `caveman providers add|remove|login|local`, and `caveman providers cloud on|off` as new commands.

Adds a status grid command and `caveman stats` listing every prompt-cache warm event.

Adds `caveman on|off` for switching modules from a signed registry backed by `modules.json` in the checksum manifest, with a module lockfile recording what is installed.

Centralizes configuration under `~/.caveman/cloud.json`, migrating older config once on upgrade.

+5 more

└──▷ **BREAKING ON UPGRADE**

! The `@caveman-ai/cli` package version jumps from 2.1.0 to 4.0.0 and binaries move from `bin-v2.1.0` to `bin-v4.0.0`; the `npx github:JuliusBrussee/caveman` installer now requires CLI `^4.0.0`, so older CLI versions will not understand what 4.0 wrote.

! Older config locations are migrated once to `~/.caveman/cloud.json` on first run; any tooling reading the previous config path will no longer find it there.

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ Take Puck with You (2026-10-13)

BLOG ↗ RANK

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

> Amp added interactive HTML prototypes rendered inline in its web app, and exposed its Puck assistant via an MCP server so external agents like ChatGPT and Claude can interact with orbs and threads.

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

01 **Puck exposed via MCP server** NEW 88

Amp now exposes Puck through an MCP server at `https://ampcode.com/mcp`, offering two tools — `puck` (interact with your Puck in Amp) and `orbs` (show an interactive view of running orbs with portal links) — so external agents like ChatGPT and Claude can reach Amp. This enables cross-agent coordination: Puck MCP can search your threads, surface earlier work, and route tasks to the correct orb from any MCP-compatible client. Setup: add `https://ampcode.com/mcp` as an MCP server in ChatGPT and sign in with your Amp account, or use Settings > Puck > Add to Claude in Amp.

Connect Amp's Puck to ChatGPT or another MCP client so you can hand off ideas or check on running orbs without leaving your current chat.

```
$ MCP endpoint: https://ampcode.com/mcp

In ChatGPT or your MCP client, add this URL as an MCP server and
sign in with your Amp account. Once connected, use the `puck`
tool to send tasks and the `orbs` tool to view running orbs.
```

Add Puck MCP directly to Claude via the Settings panel — no URL to copy.

📍 Settings › Puck › Add to Claude

— Names endpoint, tools, and exact setup steps Take Puck with You (2026-10-13)

02 **Interactive UI prototypes in web app** NEW 62

Amp can generate interactive prototype generation by writing a self-contained HTML page with CSS and JavaScript inside an `amp-view` code block, rendered inline in a sandboxed, theme-aware (light/dark) frame in the Amp web app with no network access.

The sandbox is isolated from your codebase — it uses sample data, saves nothing, and cannot reach your app, code, or data. This is distinct from Painter (generates images) and portals (shows your real app in the orb), each answering a different design question.

— Explains mechanism and scope but no command to try it `product docs`

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

CLI: Local Command Sandbox and Local Autonomous Mode: Use the local command sandbox and a

BLOG ↗ RANK

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

> Kiro's biggest addition this window is a local command sandbox and autonomous mode for containing and auto-approving agent-run shell commands, shipped in both the CLI and IDE, alongside a model fallback system, workflow loop recovery, and a wave of session-management and config changes across CLI, IDE and Web.

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

01 Local command sandbox for agent shell commands NEW

91

A new `/sandbox` command (CLI) and status-bar toggle (IDE) place agent-issued shell commands and child processes inside an OS-level sandbox, with configurable outbound network access set to blocked, limited to common development hosts, or open. Local stdio MCP servers run inside the same sandbox, and a `sandbox.json` file (global or project) with `enabled: true` applies sandbox settings to V3 headless runs, with a `--sandbox` flag available to halt the run if the sandbox cannot be applied.

Contain all agent-issued shell commands during an agentic run, blocking outbound network access to reduce risk.

📍 `/sandbox`

Persist sandbox protection across V3 headless runs by enabling it in a project-level config file; add `--sandbox` in CI to abort rather than continue unprotected.

```json
{ "enabled": true }
```

Restrict an agent session so it cannot make arbitrary outbound network calls — useful when running untrusted or third-party workflows.

📍 In the status bar, click Command sandbox › open Settings › set outbound network access to blocked or 'limited to common development hosts', then confirm the sandbox is active before starting the agent.

— Names the exact command, config file, flag and network modes

CLI: Local Command Sandbox and Local Au… | IDE: Local Command Sandbox, Start Saved…

02 **Model fallback selection** NEW 85

A fallback model picker at `Settings > Model` lets you choose a backup: if the primary model is unavailable it answers only that turn and Kiro returns to the primary next turn; if the primary refuses, the fallback takes over the rest of the session. The chat transcript now names the fallback model used and whether it applies to one turn or the rest of the conversation.

Set a fallback so an unavailable primary model never blocks your session — useful in CI or long-running agentic tasks.

> 📍 In Settings › Model, select your primary model from the list, then choose a fallback model in the fallback picker next to it.

– Exact settings path and fallback scope rules given

Web: Model Fallback, Workflow Step Mode… | IDE: Local Command Sandbox, Start Saved…

03 **Workflow settings and execution controls** NEW 80

Each workflow step's composer now has its own model picker, including effort and Thinking modes where supported, accessible when Workflows is turned on and applying only to that step; when a workflow repeat pauses at its iteration cap, Run N more accepts 1–1,000 extra iterations and Stop looping and continue moves past the loop to the next step; and uploaded `JSON` or `YAML` workflow recipes can now be edited in place under `Settings > Workflows`, with save enabled once syntax is valid and the draft preserved if a save fails.

Recover a stuck workflow loop without restarting the whole session — grant extra iterations or skip to the next step.

> 📍 When a workflow repeat pauses at its iteration cap, click Run N more, enter a whole number between 1 and 1,000, and confirm — or click Stop looping and continue to move to the next step.

Edit a workflow recipe in place to fix a syntax error without re-uploading the file.

> 📍 Go to Settings › Workflows, open the uploaded JSON or YAML recipe, make your edits, and click Save once the syntax is valid.

– Names the surface and scope but no exact navigation path Exact control names and numeric limit given Names exact settings path and save behavior

Web: Model Fallback, Workflow Step Mode…

04 **KIRO_HOME configuration unification** BREAKING 75

`KIRO_HOME`, set to an absolute path, now stores V3 global configuration, sign-ins, sessions, and agent state, supporting separate profiles such as work and personal. If `KIRO_HOME` is a relative path, a filesystem root, your home directory, or one of its parents, Kiro silently falls back to `~/.kiro` instead of using the custom value.

— Names the env var and exact fallback path, with caveat

CLI: Local Command Sandbox and Local Au…

05 **Local autonomous mode for V3 sessions** NEW 65

The `/autonomous` command lets Kiro work through tasks in local V3 sessions without per-command approval; invoking it automatically switches to the Default agent and enables the local command sandbox.

Let Kiro run through a multi-step local task autonomously without prompting for shell-command approval on each step.

📍 `/autonomous`

— Exact command and its automatic side-effects are given

CLI: Local Command Sandbox and Local Au…

06 **Session workspace and organization improvements** IMPROVED 65

The session Files tree now shows the `.agents` folder alongside session files, and repositories cloned during a session via `git`, `gh`, or `glab` appear in the session's repositories list. Opening a session or sending a message restarts its 90-day retention period, and New group in the Move to Group menu lets you create a group while moving a session.

— Names each surface but mostly passive UI behavior

Web: Model Fallback, Workflow Step Mode…

07 **Start saved Workflows from chat** NEW 60

The `/workflow-run` slash command in chat lets you type `/workflow-run`, choose a saved recipe, complete its inputs, and launch the run without leaving the chat pane.

Kick off a saved Workflow without leaving the chat pane — ideal for repeatable tasks like dependency audits or build pipelines.

📍 In the Kiro chat input, type `/workflow-run`, select the target recipe from the picker, fill in its required inputs, and press Enter to start the run.

— Exact command and step sequence given IDE: Local Command Sandbox, Start Saved…

THINNER COVERAGE BELOW

08 **Automatic image fitting to model limits** IMPROVED 35

Uploaded images are now fit to the selected model's limits at request time, refitting automatically after a model change, with a placeholder sent for models without image support.

— Explains behavior but nothing for the reader to configure

Web: Model Fallback, Workflow Step Mode…

09 **On-demand MCP tool loading in V3** IMPROVED 30

V3 now loads MCP tool definitions on demand rather than upfront, preserving more model context for active work.

— States the benefit but no mechanism or metric CLI: Local Command Sandbox and Local Au…

10 **Live session status on editor tabs** NEW 30

Chat and Workflow step tabs now show a live working or input-needed indicator, returning to their normal icon when idle.

— Describes the UI change but nothing a reader configures

IDE: Local Command Sandbox, Start Saved…

└──▷ **BREAKING ON UPGRADE**

! When `KIRO_HOME` is a relative path, a filesystem root, your home directory, or one of its parents, Kiro now falls back to `~/.kiro` — any custom `KIRO_HOME` value that matched those conditions will silently be ignored and the fallback used instead.

WAS THIS USEFUL?  

aerovato Operator Memory

Feedback ⓘ helper@v1.8.0

NOTES ↗ RANK

The self-improving context engine for coding agents.

> Operator Memory's helper v1.7.0 adds a Claude Code adapter while dropping support for the Code Puppy adapter.

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

01 **Code Puppy adapter removed** BREAKING 45

The Code Puppy adapter is removed in helper@v1.7.0; any workflow relying on it will break on upgrade.

— Clear breaking change with upgrade warning, no migration path given helper@v1.7.0

02 **Claude Code adapter** NEW 30

Helper v1.7.0 adds a new adapter for Claude Code, extending the set of coding agents Operator Memory can integrate with.

— Named only as an adapter addition, no mechanism given helper@v1.7.0

NEW INTEGRATION Kiro adapter

└──▷ **BREAKING ON UPGRADE**

! The Code Puppy adapter is removed; any workflow relying on it will break on upgrade to helper@v1.7.0.

> helper@v1.8.0 NOTES ↗ LATEST STABLE
>
> Adds a Kiro adapter to extend Operator Memory integration support.
>
> └──▷ **GET THIS VERSION** ▾
>
> ```
> $ git clone --branch helper@v1.8.0
> https://github.com/aerovato/operator-memory.git
> # already have the repo? check out this version:
> $ git checkout helper@v1.8.0
> ```

WAS THIS USEFUL? 👍 👎

TencentCloud Octop

Feedback ⓘ v1.0.2b6 NOTES ↗ RANK

A smarter, self-hosted AI assistant — multi-user, multi-agent.

> Octop v1.0.2b6 adds an Agent Mail connector, LDAP directory login for enterprise authentication, a new expert conversation mode, and native fnOS ARM64 packaging.

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

01 **Agent Mail connector** NEW 45

Adds an Agent Mail connector with authorization, mail tools, and new mail task support, letting agents interact with email as part of their workflow.

— Names the connector and sub-features but no config/API detail v1.0.2b6

02 **fnOS ARM64 native package** NEW 30

Adds an fnOS ARM64 native package and multi-architecture image support.

— Names platform/architecture but no further mechanism v1.0.2b6

03 **LDAP directory login** NEW 25

Adds LDAP directory login for enterprise authentication.

— Bare mention, no setup detail given v1.0.2b6

04 **Expert conversation mode and input bar menu** NEW 25

Adds an expert default conversation mode and a '+' menu in the input bar.

— UI additions described only briefly v1.0.2b6

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Mastra

Feedback ⓘ @mastra/core@1.77.0 NOTES ↗ RANK

Mastra is a TypeScript framework for AI applications and agents, providing workflows, agent memory, RAG, tool calling, and evals.

> Mastra 1.77.0 introduces pluggable FactorySandbox providers (Docker, E2B, Mastra Platform) for environment sandboxes, conflict-aware Observational Memory writes, W3C traceparent span linking, and run fencing across ten storage adapters, alongside new identity scopes for agent memory and a dataset-driven Experiments feature in Mastra Studio that compares up to ten runs against a baseline.

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

01 FactorySandbox provider abstraction for workspaces NEW

96

Adds the `FactorySandbox` abstract class in `@mastra/core/workspace` for pluggable sandbox providers, with `provider`, `create`, optional `settings` (zod/JSON Schema/Standard Schema), `template`, and `builds` (`.start`, `.get`, `.list`) members. The `builds` capability lets environment templates be built ahead of sessions (manual, scheduled cron, or push-triggered) via `builds.start`, `builds.get`, and `builds.list` to reduce first-session cold starts, and ships `@mastra/docker`, `@mastra/e2b`, and `@mastra/platform-workspace` as first-party implementations for Docker, E2B, and Mastra Platform. `MastraFactoryConfig.sandbox` now accepts a provider instance for settings/builds; the existing bare callback form still works but cannot expose tunable settings or prebuilt templates.

Implement a custom `FactorySandbox` provider so Factory can build environment templates ahead of sessions and plug into Mastra with a single provider instance.

typescript

```typescript
import { FactorySandbox } from '@mastra/core/workspace';
import type { FactorySandboxContext } from
'@mastra/core/workspace';
import type { PublicSchema } from '@mastra/core/schema';

interface Settings {
  region?: 'us' | 'eu';
}

class MyFactorySandbox extends FactorySandbox<Settings> {
  readonly provider = 'my-cloud';
  readonly settings: PublicSchema<Settings> = {
    type: 'object',
    properties: { region: { type: 'string', enum: ['us', 'eu'],
default: 'us' } },
    additionalProperties: false,
  };

  create(ctx: FactorySandboxContext, settings: Settings) {
    return new MyCloudSandbox({
      id: ctx.sessionId,
      sandboxId: ctx.sandboxId,
      template: this.template(ctx, settings),
      workingDirectory: ctx.workingDirectory,
      region: settings.region ?? 'us',
    });
  }

  readonly builds = {
    start: async (ctx: FactorySandboxContext, settings: Settings)
=>
      myCloud.startBuild(this.template(ctx, settings)),
    get: async (ctx: FactorySandboxContext, settings: Settings,
buildId: string) =>
      myCloud.buildStatus(buildId),
    list: async (ctx: FactorySandboxContext, settings: Settings)
=>
      myCloud.builds(this.template(ctx, settings).name),
  };
}
```

— Full class shape, provider list, and working code example `@mastra/core@1.77.0`

02 Conflict-aware writes for Observational Memory

BREAKING

88

Adds commitActiveObservations({ applied, reason }) and appendBufferedObservations({ persisted, recordId }) write-result APIs to storage, plus a `supersededBy` field on observational memory records, enabling conflict-aware and generation-safe concurrent writes. In upgraded storage adapters, `updateActiveObservations` now throws when observations were not written (superseded generation or `expectedActiveObservations` mismatch) instead of silently resolving, so `@mastra/core`, `@mastra/memory`, and storage adapters must all be upgraded together across every process writing to the same database.

Guard concurrent Observational Memory writes across processes by checking whether observations were actually applied before advancing state.

typescript

```
const memory = await storage.getStore('memory');
const result = await memory.commitActiveObservations({
  id: record.id,
  observations,
  tokenCount,
  lastObservedAt: new Date(),
  expectedActiveObservations: record.activeObservations,
});

if (!result.applied) {
  // result.reason: 'retired' (newer generation replaced record)
or 'conflict' (another writer changed it)
  // Re-read the current record and observe again instead of
silently dropping observations.
}
```

— Exact APIs and failure mode, upgrade coordination note included `@mastra/core@1.77.0`

03

Span links with W3C traceparent support NEW

85

Adds a `links` field to span creation via `getOrCreateSpan` for cross-trace references, propagated by the Datadog, Sentry, and OTel exporters. Adds parseTraceparent() and formatTraceparent() helpers in `@mastra/core/observability` for working with W3C `traceparent` values, and adds bidirectional span linking between `@mastra/mcp` client tool-call spans and MCP server request spans for cross-system agent trace navigation.

Correlate an inbound agent span to an upstream caller's trace using a W3C traceparent header, without merging the two traces.

```typescript
import { parseTraceparent, getOrCreateSpan, SpanType } from
'@mastra/core/observability';

const caller = parseTraceparent(req.headers['traceparent']);

const span = getOrCreateSpan({
  type: SpanType.GENERIC,
  name: 'handle agent request',
  links: [{ traceId: caller.traceId, spanId: caller.spanId }],
  mastra,
});
```

— Named APIs with runnable example correlating spans `@mastra/core@1.77.0`

04 **Identity scopes in agent memory** NEW 85

Memory now supports `scopes` as an alternative to discrete resource/thread objects, accepting strings in `<type>:<value>` format (e.g. `resource:user-123`, `thread:conversation-123`, `org:acme`) so a single run can carry multiple identity addresses at once. The `MASTRA_SCOPES_KEY` request context key, settable only by trusted server middleware, injects scopes server-side without client exposure; tools, processors, and subagents receive the run's non-resource, non-thread scopes (such as `org:acme`), while each nested agent call independently selects its own `resource:` and `thread:` scope.

— Full mechanism and exact key names, no code sample given `product docs`

05 **Workflow HTTP routes no longer start internal workflows** 81

BREAKING

`@mastra/server` workflow HTTP routes no longer allow starting or resuming internal workflows (returns 404); durable agents must now be invoked via agent routes such as `POST /agents/:agentId/stream`.

— Exact endpoint and replacement route given `@mastra/core@1.77.0`

06 **Message authorship, thread transfer, and file attachments in AgentController**

NEW 78

Adds an `author` parameter to `sendMessage`, `queueMessage`, and `steer` on AgentController sessions, readable back via `getMessageAuthor`. Adds thread.transferOwnership({ toOwnerId }) for transferable thread ownership across distributed runs, and adds `files` attachment support to `steer` and `followUp` (and

their `/steer` and `/follow-up` HTTP routes), matching the existing `sendMessage` capability.

— Names exact methods and routes but no usage example `@mastra/core@1.77.0`

07 **Experiments feature in Mastra Studio** NEW 70

Mastra Studio adds an Experiments feature that runs every item in a dataset against an agent, workflow, or scorer, then shows each item's input, output, status, and individual score breakdowns in one place. Up to ten experiments can be compared side by side against a baseline to measure the impact of prompt, model, or code changes; the `@mastra/client-js` and `@mastra/react` compare-experiments APIs now take `experimentIds` (with an optional `baselineId`) instead of the deprecated `experimentIdA` / `experimentIdB` parameters.

— Names API params and limits but no direct runnable example `@mastra/core@1.77.0`

08 **Run fencing across storage adapters** NEW 68

Adds run fencing across Cloudflare DO/D1, Convex, DynamoDB, DSQL, MongoDB, MSSQL, MySQL, OracleDB, Spanner, and Redis/Upstash/Valkey storage adapters so a recovered durable execution cannot overwrite threads, messages, or workflow state from a prior owner.

— Lists every adapter covered but no config/flag to act on `@mastra/core@1.77.0`

09 **Controlled props for Dialog and AlertDialog components** 67

BREAKING

In `@mastra/playground-ui`, `Dialog` and `AlertDialog` components no longer accept handle/render-style props; they are now controlled via `open` and `onOpenChange`.

— Exact prop names given but no code migration sample `@mastra/core@1.77.0`

10 **Deprecated SessionRecord fields and HarnessStorage methods** 61

DEPRECATED

Deprecates `SessionRecord` fields `modeId`, `modelId`, `state`, and `pending`; `HarnessStorage` methods `appendPendingItem`, `updatePendingItem`, and `removePendingItem`; and SessionSuspensions.resolveToolCallId().

— Names every deprecated field and method, no migration path given `@mastra/core@1.77.0`

THINNER COVERAGE BELOW

11 **Custom pricing for gateway provider configs** NEW 57

Adds a `pricing` field to gateway provider configs so custom or reseller models can report input/output costs in USD per 1M tokens (same keys as models.dev) for cost estimation.

— Names the field and units but no example of use `@mastra/core@1.77.0`

└─▷ BREAKING ON UPGRADE

! `MastraFactoryConfig.sandbox` now accepts a provider-instance (`FactorySandbox`) for settings/builds; the existing bare callback form still works but cannot expose tunable settings or prebuilt templates.

! `@mastra/server` workflow HTTP routes no longer allow starting or resuming internal workflows (returns 404); durable agents must be invoked via agent routes such as `POST /agents/:agentId/stream` .

! `SessionRecord` fields `modeId` , `modelId` , `state` , and `pending` are deprecated; `HarnessStorage` methods `appendPendingItem` , `updatePendingItem` , and `removePendingItem` are deprecated; SessionSuspensions.resolveToolCallId() is deprecated.

! `@mastra/playground-ui` : `Dialog` and `AlertDialog` components no longer accept handle/render-style props; they are now controlled via `open` and `onOpenChange` .

! `@mastra/client-js` and `@mastra/react` compare-experiments APIs deprecate `experimentIdA` and `experimentIdB` in favor of `experimentIds` (with optional `baselineId`).

! In upgraded storage adapters, `updateActiveObservations` now throws when observations were not written (superseded generation or `expectedActiveObservations` mismatch) instead of silently resolving; `@mastra/core` , `@mastra/memory` , and storage adapters must all be upgraded together across every process writing to the same database.

WAS THIS USEFUL? 👍 👎

CrewAI

Feedback ⓘ 1.15.28 NOTES ↗ RANK

Open-source Python framework for building multi-agent workflows using role-based AI agent Crews and event-driven Flows.

> CrewAI 1.15.28 adds a diagnostic eval command for crew runs and switches context-window lookups to the LiteLLM catalog.

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

01 `crewai eval` command for run diagnostics NEW 70

A new `crewai eval` command reports where a stopped model broke during a crew run, letting developers pinpoint the exact step that caused a failure without re-running the full crew.

Diagnose exactly which step caused a model to stop mid-run, so you can pinpoint prompt or context issues without re-running the full crew.

```
$ crewai eval
```

— Names exact runnable command with clear purpose. 1.15.28

THINNER COVERAGE BELOW

02 LiteLLM catalog for context-window lookups IMPROVED 35

Context-window lookups now use the LiteLLM catalog instead of a hardcoded table, replacing a static internal mapping with a maintained external source.

— Explains mechanism change but no actionable steps given. 1.15.28

WAS THIS USEFUL? 👍 👎

LiveKit Agents

Feedback ⓘ livekit-agents@1.8.7 NOTES ↗

LiveKit Agents builds realtime voice and video AI agents on LiveKit's WebRTC infrastructure, with turn detection, interruption handling, and telephony.

Adds E2E latency metrics for realtime agents, `CacheBreakpoint`/Instructions(dynamic=...) for OpenAI prompt caching, `preload_modules` for forkserver workers, and `reduce_silence` for Soniox TTS.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch livekit-agents@1.8.7
https://github.com/livekit/agents.git
# already have the repo? check out this version:
$ git checkout livekit-agents@1.8.7
```

Adds event.item.metrics.get('e2e_latency') on the `conversation_item_added` event to measure realtime reply latency from last detected user speech to first assistant playback (requires local VAD).

Adds `CacheBreakpoint` content marker for OpenAI prompt caching, automatically inserted at the end of the instructions message each turn to maximize cache hits.

Adds Instructions(dynamic=...) for per-call dynamic text placed behind a cache breakpoint, keeping static prefixes cached across turns.

Adds `preload_modules` to forkserver preload configuration, allowing modules to be loaded before workers fork.

Adds `reduce_silence` option to the Soniox TTS provider.

+1 more

NEW INTEGRATION OpenAI gpt-live-transcribe STT via inference provider

NEW INTEGRATION Fish Audio drama-3 TTS model support

NEW INTEGRATION Gemini 4 Argon model support

UPDATED INTEGRATION Anthropic Claude Opus 5, Opus 5.5, Sonnet 5.5 model support

UPDATED INTEGRATION language_hint added to DeepgramFluxOptions

└──▷ BREAKING ON UPGRADE

! Telemetry: fallback attributes `lk.fallback.label` and `lk.fallback.index` move to adapter request spans; existing trace queries relying on the old hierarchy must be updated.

! Anthropic: custom SDK HTTP clients must use `httpx2` instead of `httpx`.

! OpenTelemetry: the Events API has been removed; replace usages with LogRecord(event_name=...); custom `Logger` subclasses must now implement enabled().

! Dependencies: now requires `openai>=3.19,<4`, `mcp>=2.2,<3`, `websockets>=17.1,<17.2`, OpenTelemetry `>=1.45,<1.46`, Anthropic `>=1.8,<2`, and `onnxruntime>=1.30,<1.31` for turn detection.

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@7.0.140 NOTES ↗ RANK

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

> Vercel AI SDK 7.0.140 adds configurable concurrency for tool call execution.

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

01 Configurable tool call execution concurrency IMPROVED

35

Callers can now control how many tool calls run in parallel during execution.

— No flag or config name given, just prose description. ai@7.0.140

WAS THIS USEFUL? 👍 👎

Skyvern

Feedback ⓘ v1.0.56 NOTES ↗ RANK

Skyvern automates browser-based workflows with vision models and agents, completing form-driven tasks on sites it has not seen before.

> Skyvern v1.0.56 introduces serial workflow run groups for batch execution, an `ENABLE_URL_VALIDATION` kill switch, audit logging across credentials/folders/schedules/tags/scripts, and a new Claude Haiku 5.5 model option, alongside smaller improvements to Copilot credentials, file parsing, and error classification.

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

01 ENABLE_URL_VALIDATION kill switch for SSRF checks NEW

80

Adds `ENABLE_URL_VALIDATION` kill switch to disable SSRF host checks when needed, settable in a `.env` file as `ENABLE_URL_VALIDATION=false`.

Disable SSRF host validation in a self-hosted deployment where internal URLs are intentional targets.

```env
ENABLE_URL_VALIDATION=false
```

— runnable config example included v1.0.56

02 Serial workflow run groups NEW

65

Adds serial workflow run groups that submit, recover, and report up to 25 runs as one operation.

Inspect the resolved run type and timeline for a specific run via the API.

```
$ curl -X GET 'https://<your-skyvern-host>/v1/runs/<run-id>/timeline' \
  -H 'x-api-key: <your-api-key>'
```

— includes API example to inspect run timeline v1.0.56

03 **Claude Haiku 5.5 model option** NEW — 60

Adds Claude Haiku 5.5 as a selectable `LLM_NAME` variant for v1, with an Amazon Bedrock fallback.

— no example command for setting LLM_NAME v1.0.56

THINNER COVERAGE BELOW

04 **original_filename template variable for downloads** NEW — 55

Adds `{{ original_filename }}` template variable so a download can keep the site's own filename.

— no usage example provided v1.0.56

05 **Longer file-parser LLM deadline** IMPROVED — 55

Raises the file-parser LLM call deadline from 30 s to 250 s, unblocking large-file parsing use cases.

— concrete before/after values, no config key v1.0.56

06 **Audit logging for core resource writes** NEW — 50

Adds audit logging for credential, folder, schedule, tag, and script writes, with a cap on how long an audit write can hold a request.

— no named config key for the cap v1.0.56

07 **Copilot credential improvements** IMPROVED — 50

Improves Copilot credentials: account-management scope is added and kept in scope even when earlier conversation turns drift, and users can now sign in directly from the Copilot credential card.

— UI path given, no exact steps v1.0.56

08 **Bot protection error classification** IMPROVED — 45

Classifies site rate-limit or 403 pages as bot protection rather than AI model errors.

— no config or command to invoke v1.0.56

09 **Skyvern 1.0 engine in picker** NEW — 35

Offers the Skyvern 1.0 engine in every workflow's engine picker.

— thin description, just names the picker v1.0.56

10 **Record Browser feedback prompt** NEW 35

Asks for helpful/not-helpful feedback after Record Browser generates steps.

— thin UI description v1.0.56

WAS THIS USEFUL? 👍 👎

Nous Research Hermes

Feedback ⓘ v0.22.0 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 v0.22.0's biggest changes are on-device local model hosting via llama.cpp and a unified Connectors system that replaces the MCP tab, alongside a switch to stable SemVer release channels (with an opt-in main track), a new hermes pm package manager, guided first-launch onboarding, natural-language theming, and password-manager sign-in for the browser agent.

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

01 **On-device local model hosting via llama.cpp** NEW 95

The `hermes model` → Local models terminal flow and Settings → Providers → Local Models desktop path install and run llama.cpp-backed open models entirely on-device with no API key, across CPU, CUDA, Metal, Vulkan, and ROCm backends. The local model catalog includes Qwen3.8 27B, Qwen3.8 Flash Next (with MTP speculative head), Qwen3.6 35B-A3B, DeepSeek V4 Flash, and any GGUF from Hugging Face.

Browse available on-device models and start the llama.cpp runtime to run inference without an API key.

```
$ hermes model
```

— Names command, UI path, backends, and specific model catalog. v0.22.0

02 **Stable release channels with opt-in main track** BREAKING

90

Source installs now track tested SemVer releases by default instead of every commit on `main`; the `hermes update --set-channel main` CLI flag and a channel switch in Desktop Settings → About let users opt back into the bleeding-edge `main` build track.

Switch a source install back to tracking every commit on `main` after the default changed to stable SemVer releases.

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

— Names exact flag, UI path, and before/after behaviour. v0.22.0

03 **Unified Connectors for hosted apps and MCP servers** BREAKING

90

The `hermes connectors` CLI command and `/connectors` slash command manage hosted-app integrations (Gmail, Linear, Notion) and named multi-account connections (e.g. a `work` and `home` Gmail) without a Nous account. A new Connectors desktop page consolidates hosted apps and MCP servers into a single list, replacing the previous MCP tab; existing MCP server configurations are merged into this list.

Connect a work Gmail account and a personal Gmail account so the agent can act in both in the same session.

```
$ hermes connectors
```

— Names CLI command, slash command, UI page, and replaced tab. v0.22.0

04 **Guided first-launch onboarding chat** NEW

85

A first-launch guided setup chat powered by the free `nous/welcome` trial model runs once per OS user with no API key required, and can be replayed via Settings → Run onboarding again. The `onboarding.run` config flag in the root config controls whether first-run onboarding executes, backed by `onboarding.start` / `onboarding.skip` gateway RPCs.

— Names config flag, RPCs, model, and UI path to replay. v0.22.0

05 **Unified `hermes pm` package manager** NEW

75

The new `hermes pm` subcommand owns every pinned, hash-checked tool and Python environment, building new environments atomically before switching so a failed update cannot leave a half-applied state.

— Explains mechanism but gives no runnable example. v0.22.0

06 **Password-manager sign-in for browser agent** NEW 70

The browser agent can sign in using 1Password, Bitwarden, or Hermes's own encrypted vault, with the model never seeing credentials; one-time codes fill automatically and card/address fills require confirmation.

— Describes mechanism and scope but no command or config key. v0.22.0

THINNER COVERAGE BELOW

07 **Natural-language theme changes via `set_theme`** NEW 55

The new `set_theme` agent tool lets users change the desktop theme using natural language (e.g. 'make it violet, dark mode') from any desktop chat.

— Names the tool and gives an example phrase, but thin otherwise. v0.22.0

└──▷ **BREAKING ON UPGRADE**

! Source installs now track tested SemVer releases by default instead of every commit on `main`; use `hermes update --set-channel main` to restore the previous behaviour.

! The desktop MCP tab is replaced by the new `Connectors` page; existing MCP server configurations are merged into the unified Connectors list.

WAS THIS USEFUL?  

PydanticAI

Feedback ⓘ v2.56.0 NOTES ↗ RANK

PydanticAI is a framework that builds type-safe Python agents with dependency injection, model integrations, tools, and structured outputs.

> PydanticAI agents can now run durably across nine workflow engines, and gained new classes for managing tool output during agent runs.

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

01 **Durable agent execution across nine workflow engines**

NEW INTEGRATION NEW 53

PydanticAI agents can now run on nine workflow engines: Temporal, DBOS, Prefect, Restate, AWS Lambda, Kitaru, Airflow, Absurd, and pgtask.

— Names nine engines but no setup or mechanism detail. product docs

02 **Tool output management controls** NEW 52

Adds ClearToolResults(), ToolOutputLimits(), and RepairToolArguments() controls for managing tool output in agent runs.

— Names three classes but not their exact behavior. product docs

NEW SUPPORTED PLATFORM Python 3.15 support

NEW SUPPORTED PLATFORM Gemini 4 Argon model support on GoogleModel

v2.56.0 NOTES ↗ LATEST STABLE

Adds `SmolSandbox` microVM workspace, `base_url` for `ZaiProvider`, enhanced Skills layering, and opt-in `BrowserUse` telemetry with host-model default.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.56.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.56.0
```

Adds `SmolSandbox`: a workspace that runs harness capabilities inside a local `smolvm` microVM.

Adds `base_url` argument to `ZaiProvider` to point the provider at a custom endpoint.

Lets `Skills` layer project and personal libraries, name each skill's directory, and render a skill for a host command.

Removes `WarnNearLimits` from the harness `Coder` capability's default bundle.

Makes `BrowserUse` run on the host model by default and makes browser-use telemetry opt-in.

└─▷ **BREAKING ON UPGRADE**

! `BrowserUse` now runs on the host model by default (previously used a separate model path); browser-use telemetry is now opt-in rather than enabled by default — existing setups relying on automatic telemetry or the prior model routing will behave differently after upgrade.

WAS THIS USEFUL? 👍 👎

LangChain LangGraph

Feedback ⓘ `cli==0.4.34` NOTES ↗

RANK

Low-level orchestration framework for building, managing, and deploying long-running, stateful AI agents.

> LangGraph CLI adds a JSON output option for listing deployment listeners, enabling scripted/machine-readable consumption.

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

01 **JSON output for deploy listeners list** NEW 70

The `deploy listeners list` command gains a `--json` flag to produce machine-readable JSON output, suitable for piping into tools like `jq`.

Pipe deployment listener data into jq or other tooling by requesting JSON output from the CLI.

```
$ langgraph deploy listeners list --json | jq '.[].name'
```

— Names exact flag and command, includes runnable example `cli==0.4.34`

WAS THIS USEFUL? 👍 👎

Stacklok Mecatl

Feedback ⓘ v0.0.48 NOTES ↗ RANK

Mecatl is an open-source agent harness for running the same provider-agnostic agent loop locally, remotely, or on Kubernetes, with durable state, permissions, and gRPC/HTTP-SSE APIs.

> Mecatl v0.0.48 adds host-level microVM guest network egress controls and default placement configuration, encrypts ToolHive credential storage, hardens OIDC/Kubernetes authentication, and rounds out Mecatl Studio with session inspection panels, a minimap, and a full schedule management workspace.

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

01 MicroVM guest network egress controls and default placement

NEW 95

Adds the `execution.microvm.guest_egress` config key, along with `--microvm-guest-egress=deny-all|allowlist` and repeatable `--microvm-guest-allow=HOST:PORT/tcp|udp` flags on `mecated serve`, letting the host operator control guest network egress policy. Also adds `execution: default_placement: microvm-local` in `~/.config/mecatl/settings.yaml` to route sessions into local microVM environments by default.

Lock down microVM guest network to an explicit allowlist — useful when you want agent sessions to reach only your internal API and no other external hosts.

```
$ mecated serve --headless --default-placement microvm-local --microvm-guest-egress=allowlist --microvm-guest-allow=api.internal.example.com:443/tcp
```

Set the default session placement to microVM isolation for an operator-managed deployment via config, so all sessions use isolated environments without per-client configuration.

```yaml
execution:
  default_placement: microvm-local
  microvm:
    guest_egress: deny-all
```

— Names exact flags, config keys and includes runnable examples v0.0.48

02 WebSearch provider configuration in operator settings NEW

60

Adds WebSearch configuration in operator settings, supporting Exa and Brave API keys read from `auth.yaml`.

— Names providers and config file but no key names v0.0.48

THINNER COVERAGE BELOW

03 **OIDC and Kubernetes token authentication hardening** IMPROVED

45

Hardens OIDC authentication to require HTTPS and adds validation of Kubernetes tokens.

— States the change but no config or migration detail v0.0.48

04 **Provider-scoped model aliases and delegation selectors** NEW

45

Adds provider-aware model aliases and delegation selectors so model routing can be scoped per provider.

— Explains behaviour but no config syntax shown v0.0.48

05 **Session inspection, minimap, and schedule management in Mecatl Studio**

NEW

45

Adds session inspection panels and a minimap, and completes the schedule management workspace, in Mecatl Studio.

— Names UI additions but no navigation path given v0.0.48

06 **API compatibility check between mecatui and server** NEW

40

Enforces an API compatibility check between `mecatui` and the server before startup.

— Names the two components but no error or flag detail v0.0.48

07 **MCP client secrets from bounded files** IMPROVED

35

MCP client secrets can now be read from bounded files instead of only inline values.

— No file path, key, or flag named v0.0.48

08 **Execution templates with platform-managed identity** NEW

35

Adds execution templates with platform-managed identity for running agent workloads.

— No config keys, flags, or usage path given v0.0.48

09 **Encrypted ToolHive credential storage in broker** IMPROVED

30

Encrypts ToolHive credential storage in the broker.

— Names ToolHive but no mechanism or key detail v0.0.48

10 **Skill file access via client API** NEW

30

Adds skill file access through the client API.

— No endpoint or method named for the API v0.0.48

WAS THIS USEFUL? 👍 👎

Google ADK

Feedback ⓘ v2.12.0 NOTES ↗ RANK

The Agent Development Kit is Google's code-first Python toolkit for building, evaluating, and deploying agents, with built-in tools, sessions, and deployment to Vertex AI.

> ADK v2.12.0 adds tool-output filtering via `after_tool_callback`, human-approval gating via `require_confirmation` on workflow nodes, pre-seeded CLI session state via `--state_file`, custom service accounts for Agent Engine deployment, and expands sandbox, task, skill, and event-handling support alongside new Python 3.15 support.

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

01 after_tool_callback for tool output filtering NEW

80

Adds `after_tool_callback` to intercept and filter tool outputs before they are returned to the model, letting an agent redact or transform tool results — such as masking PII — before the model sees them.

Screen or transform tool output before it reaches the model — e.g. redact PII or truncate large payloads.

python

```python
from google.adk import Agent

def redact_tool_output(tool_result):
    # Inspect or mutate the tool result before the model sees it
    if 'ssn' in str(tool_result).lower():
        return {'output': '[REDACTED]'}
    return tool_result

root_agent = Agent(
    name='my_agent',
    model='gemini-3.5-flash',
    instruction='You are a helpful assistant.',
    after_tool_callback=redact_tool_output,
)
```

— Named callback with a runnable code example of redaction. v2.12.0

02 require_confirmation for workflow node approval NEW

80

Adds `require_confirmation` to `@node` and `FunctionNode` to require human approval before a workflow node executes, pausing the node until confirmation is given.

Gate a sensitive workflow node behind a human approval step before it executes.

```python
from google.adk.workflow import node

@node(require_confirmation=True)
def delete_records(state):
    # This step will pause and wait for human confirmation before
running
    ...
```

— Named flag with a runnable example gating a sensitive step. v2.12.0

---

03 **--state_file option for adk run** NEW 75

Adds `--state_file` option to `adk run` for loading pre-seeded session state from a file, so a CLI session can resume with initial context already set instead of re-entering it manually.

Resume an interactive CLI session with pre-seeded state so you do not have to re-enter context manually.

```
$ adk run path/to/my_agent --state_file
path/to/initial_state.json
```

— Named CLI flag with an exact runnable command. v2.12.0

THINNER COVERAGE BELOW

04 **Skill security and telemetry in AgentCardBuilder** IMPROVED 50

Adds skill-level security support to `AgentCardBuilder`, persists skill-script outputs as artifacts, and names skill tool spans after the skill they touch in telemetry.

— Three related additions described but no usage example. v2.12.0

---

05 **Tool authentication and credential resume in _ToolNode** NEW 45

Supports tool authentication requests and credential resume in `_ToolNode`.

— Named internal class but no example or flow detail. v2.12.0

---

06 **Schema callback hooks in BigQueryLoggerConfig** NEW 40

Adds `on_schema_error` and `on_schema_ready` callback hooks to `BigQueryLoggerConfig`.

— Named hooks only, no behavior detail or example. v2.12.0

07 **Custom service accounts in Agent Engine deployment** NEW

35

Supports custom service accounts when deploying agents to Agent Engine.

— Named capability but no mechanism or example given. v2.12.0

08 **Richer state preserved in ADK events** IMPROVED

35

Preserves model thinking parts, client tool responses, and sub-agent hierarchy in ADK events.

— States what's preserved but not how to access it. v2.12.0

09 **Sandbox lifecycle helpers and template reuse** IMPROVED

35

Adds sandbox lifecycle helpers to the sandbox manager and finds or creates sandbox templates per container config.

— Named module addition but no example or API detail. v2.12.0

10 **Sub-agent delegation and resume in task mode** NEW

30

Supports task sub-agent delegation and resume in task-mode wrapper.

— Brief prose description, no concrete surface named. v2.12.0

11 **Python 3.15 support** NEW

15

Adds support for Python 3.15.

— Mentioned only in summary with no further detail. v2.12.0

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

Composio

Feedback ⓘ @composio/cli@0.4.4-beta.426 NOTES ↗ RANK

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

> Composio's CLI 0.4.4-beta.426 adds a session-level control for proxy-based tool execution.

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

01 **Proxy execution control for sessions** NEW 55

Session create and update calls now accept a `proxyExecute` parameter, enabling proxy-based tool execution to be controlled at the session level.

— Names exact parameter but lacks behavior detail or example @composio/cli@0.4.4-beta.426

WAS THIS USEFUL? 👍 👎

gram

Feedback ⓘ server@2.23.0 NOTES ↗ RANK

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

> Gram's server and dashboard releases this window add pre-deploy OpenAPI diffing and review, a remote-session identity-impact preview API, plain-language risk policy drafting, and an Okta-gated identity chaining control for remote MCP servers.

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

01 Pre-deploy OpenAPI diff and review NEW

91

Adds `deployments.evaluateOpenAPIv3` to compare an uploaded OpenAPI document against the deployed version without deploying it, and `deployments.evaluation` to compare a deployment with the one before it. Uploading a new OpenAPI source version now also triggers a pre-deploy review listing breaking changes, removed tools, new required inputs, and new or removed credentials, along with which MCP servers use the source and how many tools are affected.

Validate an updated OpenAPI spec against the live deployment to catch breaking changes before any rollout.

```text
deployments.evaluateOpenAPIv3({ document: '<uploaded-document-id>' })
```

Review what changed between the current deployment and its predecessor without touching production.

```text
deployments.evaluation({ deployment_id: '<deployment-id>' })
```

— Named endpoints, review contents, and runnable usage examples. server@2.23.0

02 Identity chaining control for remote MCP servers NEW

85

Behind the `okta-connections` flag, adds an Identity chaining control for remote MCP servers, with `prepareEMA` to enable and `unlinkEMA` to disable it, a Requested scopes picker defaulting to the linked client's scopes minus sign-in scopes (`openid`, `profile`, `email`, `offline_access`, `phone`, `address`), and blocking enablement for public clients.

— Feature flag, function names, and scope defaults all named. `dashboard@0.137.0`

03 **Remote session identity-impact preview API** NEW 74

Adds `remoteSessions.getServerIdentityImpact` to preview which MCP servers and gateways a client binding change on a user session issuer would affect, classifying each server as `repoint`, `clear`, or `resignin` and each gateway as `client_removed`.

— Named endpoint and classification values but no usage example. `server@2.23.0`

04 **MCP metadata API parameter additions** IMPROVED 60

Adds `mcp_server_id` as an accepted parameter to `setMcpMetadata` (alongside `toolset_slug`), and makes `mcp_slug` optional in `exportMcpMetadata`.

— Named parameters and endpoints but minimal purpose given. `server@2.23.0`

THINNER COVERAGE BELOW

05 **Plain-language risk policy drafting** NEW 50

Enables starting a new risk policy from a use case catalog or a plain-language description, with the Platform MCP able to list use cases, suggest a policy from a description, and create a policy from a use case.

— Describes workflow but no concrete endpoint or command. `server@2.23.0`

06 **Toolset slug alias in onboarding agent** IMPROVED 45

Adds `toolset_slug` as an accepted parameter to the onboarding agent's `attach_mcp_server`, keeping `attach_toolset` as an alias.

— Named parameter alias but minimal surrounding context. `dashboard@0.137.0`

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Serving & Optimization

FlashAttention

Feedback ⓘ fa4-v4.0.0.beta35 NOTES ↗ RANK

FlashAttention implements fast, memory-efficient exact attention kernels used by most current transformer training and inference stacks.

> FlashAttention-4 beta35 adds automatic num_splits tuning for CuTe kernels on Blackwell (SM120) GPUs, removing a manual tuning step.

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

01 **Automatic num_splits tuning on SM120 Blackwell** IMPROVED

50

FlashAttention-4 beta35 adds automatic `num_splits` selection for CuTe kernels on SM120 (Blackwell) GPUs, removing the need to manually tune this parameter.

— Names the exact parameter and GPU target but no benchmark or mechanism detail.

fa4-v4.0.0.beta35

WAS THIS USEFUL? 👍 👎

◆ Agentic Infrastructure

OpenAI TypeScript SDK

Feedback ⓘ v7.33.0 NOTES ↗

RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> OpenAI's TypeScript SDK v7.33.0 expands API coverage with browser and computer toolsets, new storage and reporting surfaces, and refined image/moderation detail, alongside beta session spending controls.

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

01 **Beta session spending limits and error code** NEW 40

Adds a `billing_not_active` session error code and beta session spending limits to the API.

— Names the error code but no limit values or usage shown v7.33.0

02 **Image tool and moderation detail exposure** IMPROVED 40

Adds image URLs and detailed safety alert explanations, and exposes image tool error and moderation details in the API.

— Describes what's exposed but no field names or endpoints v7.33.0

03 **Browser and computer toolsets (initial phase)** NEW 40

Adds browser and computer toolsets to the API, described as an initial phase of support.

— Notable new capability but no tool names or usage shown v7.33.0

04 **New Live voice: shida** NEW 35

Adds `shida` to the named Live voices available in the API.

— Names the exact voice value but no usage example v7.33.0

05 **OCI external storage provider** NEW 25

Adds OCI external-storage provider support to the API.

— Bare mention, no config keys or usage detail v7.33.0

06 **Organization Decisions usage reporting** NEW 25

Adds organization Decisions usage reporting to the API.

— Bare mention with no endpoint or field names v7.33.0

07 **Vault updates and metadata filters** NEW 25

Adds vault updates and metadata filters to the API.

— No field names or endpoint given for vault changes v7.33.0

08 **Project API key expiry updates** NEW 25

Adds support for updating project API key expiry.

— No endpoint or parameter name specified v7.33.0

09 **Beta pagination and parameter alignment** IMPROVED 25

Aligns beta pagination limits and non-null parameters across the API.

— General alignment noted without specific endpoints or values v7.33.0

WAS THIS USEFUL?  

OpenAI Python SDK

Feedback ⓘ v3.31.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI Python SDK v3.31.0 introduces browser and computer toolsets for agentic automation, while v3.30.0 expands the underlying API surface with new storage, usage-reporting, spending-limit, and image/safety response fields.

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

01 **Browser and computer toolsets for agents** NEW 40

Adds `browser` and `computer` toolsets to the SDK, enabling agentic browser control and computer interaction capabilities.

– Names the toolsets but no usage detail v3.31.0

02 **Image API error, moderation and safety details** IMPROVED 35

Exposes image tool error and moderation details in API responses, and supports image URLs plus detailed safety alert explanations in API responses.

– Groups two related response-field additions to image API v3.30.0

03 **OCI external-storage provider support** NEW 30

Adds OCI (Oracle Cloud Infrastructure) external-storage provider support to the API.

– Names the provider, no mechanism given v3.30.0

04 **Project API key expiry update** NEW 30

Supports updating project API key expiry via the API.

– Clear capability, no endpoint name given v3.30.0

05 **Organization Decisions usage reporting** NEW 25

Adds organization Decisions usage reporting to the API.

– Bare description of a new reporting capability v3.30.0

06 **Beta session spending limits** NEW 25

Adds beta session spending limits to the API.

– Named as beta feature but no config detail v3.30.0

WAS THIS USEFUL?  

Tracer Cloud OpenSRE

Feedback ⓘ v0.1.2026.10.14 NOTES ↗

RANK

Build your own AI SRE agents. The open source toolkit for the AI era.

> OpenSRE's two-week stream centered on scheduling and agent-loop infrastructure — templates, listing, and new `scheduled_task_reported` / `scheduled_tasks_registered` analytics events — alongside a tightened GitHub/Slack integration model gated behind OpenSRE app connections, new skills for merge-conflict resolution and local-repo context, and a steady run of CLI additions including a `/tools` browser, `/rename` command, and guided input forms.

└──▷ **WHAT SHIPPED · 21 FEATURES** most completely described first ? what's the number?

01 **Scheduled loop templates, listing and analytics** NEW

85

Ships short loop templates that every scheduled tick runs, keeps a local record of what each scheduled run sent, sends delivered loop reports as `scheduled_task_reported` events, and reports saved tasks as a `scheduled_tasks_registered` event. Lists scheduled loops by what they do rather than when they fire, shows each scheduled loop its previous runs and the actions they took, and identifies scheduled runs in analytics while recording who created each loop.

— Names concrete event types with full before/after behaviour v0.1.2026.10.5

02 **Expanded action agent context sources** NEW

70

Gives the action agent new context: the `analyzing-local-repositories` skill for reading insights from local repositories when GitHub is unavailable, the active repository's `AGENTS.md` loaded into its prompt, and the current date and time.

— Names sources, mechanism, and fallback condition v0.1.2026.10.5

03 **`/rename` command for sessions** NEW

70

Adds the `/rename` command for sessions, letting users rename active sessions from the CLI.

— Exact runnable command with clear purpose v0.1.2026.10.1

04 **Onboarding flow improvements for GitHub setup** IMPROVED

65

Invites a repo link in the empty shell prompt; gates GitHub demos on a usable token and resumes after setup during onboarding; offers outcomes before deployment choices;

records both onboarding answers when automation opens a follow-up question; asks for the demo repository before any gateway or scan work begins; and creates the demo branch automatically when `GitHub` reports a missing reference.

— Enumerates onboarding steps without exact UI screens v0.1.2026.10.4 v0.1.2026.10.3

05 **Guided editable input forms across shell commands** NEW

60

Adds guided input to the `guide /loops` shell command; adds an editable condition form in the shell to guide goal set input; shares editable input forms across work and cron in the REPL for consistent handling between execution contexts; and adds guided work-add input with editable detail rows.

— Names the shell surfaces but not exact workflow steps v0.1.2026.10.14 v0.1.2026.10.13 v0.1.2026.10.11 v0.1.2026.10.10

06 **OpenSRE app integrations for GitHub and Slack**

UPDATED INTEGRATION NEW

60

Requires OpenSRE app connections for all GitHub agent work, enforcing a consistent auth model across agent-driven GitHub operations; adds personal GitHub integrations so individual users can connect their own accounts; connects Slack in the OpenSRE app before the Slack demo flow; and uses OpenSRE app integrations in the signed-in shell and CLI.

— Names GitHub, Slack, shell and CLI surfaces without exact config v0.1.2026.10.9 v0.1.2026.10.4 v0.1.2026.10.3

07 **Skills-based agent loop execution** NEW

60

Supports running stateless agent loops directly on installed skills, including a new `fixing-github-merge-conflicts` skill that handles conflict loops automatically, and keeps conflict sweeps quiet when running an agent loop from a skill.

— Describes capability without exact invocation syntax Names the skill but not its invocation steps

v0.1.2026.10.6 v0.1.2026.10.5

08 **CLI browsers for integrations, MCP and tools** NEW

60

Enables opening integrations and MCP directly into connection browsers from the CLI, and adds a compact `/tools` browser with a fixed preview to the CLI.

— Names CLI browser surfaces with specific `/tools` command v0.1.2026.10.6 v0.1.2026.10.5

THINNER COVERAGE BELOW

09 **Skills release signing trust and auto-refresh** NEW

55

Trusts the production skills release signing key for skill verification, and pulls live skills releases automatically every five minutes.

– Gives refresh interval and trust mechanism, no config key v0.1.2026.10.4 v0.1.2026.10.3

10 **Subprocess timeout raised to 240 seconds** IMPROVED 50

Raises the subprocess limit to 240s, enabling longer-running local tasks.

– Gives an exact limit but no config key v0.1.2026.10.1

11 **Unified responsive UI for records and command lists** IMPROVED 45

Unifies responsive command lists and reduces output noise in the UI, unifies record column spacing and metadata hierarchy, and unifies cron, work, loops, and tasks with responsive record output for a consistent view across job types.

– Describes UI unification without exact UI path v0.1.2026.10.10 v0.1.2026.10.9

12 **GitHub connection status and selection in shell** IMPROVED 45

Reports GitHub connection snapshots at shell startup and on integration changes, and selects GitHub connections across shell sessions and cloud prompts.

– Names the behaviour but not an exact command v0.1.2026.10.4 v0.1.2026.10.3

13 **Conversation replay with token-based compaction** NEW 45

Replays earlier conversation turns with their tool results, compacting by token count.

– Names the compaction basis but no limit figures v0.1.2026.10.5

14 **Parallel remote prompts with safer gateway turns** IMPROVED 45

Runs remote prompts in parallel and makes 8 turns per task safe via gateway improvements.

– Gives a concrete turn limit but no config path v0.1.2026.10.4

15 **Pipedream integrations via secure webapp proxy** NEW INTEGRATION IMPROVED 40

Calls Pipedream integrations through the secure webapp proxy instead of directly.

– States the routing change without further mechanism v0.1.2026.10.7

16 **Progressive agent tool disclosure** NEW 35

Progressively discloses agent tools, surfacing only relevant tools contextually rather than all at once.

– Describes behaviour with no mechanism or UI path v0.1.2026.10.9

17 **Session memory ranking and extraction** NEW 35

Ranks memories by relevance, fences demo runs, and extracts memories from the session log.

– States mechanism briefly with no config or command v0.1.2026.10.5

18 **Provenance capture installs OpenSRE in container** IMPROVED 35

Installs OpenSRE inside the provenance container before recording, enabling provenance capture without a pre-existing installation.

– Describes mechanism without a runnable command v0.1.2026.10.1

19 **Sequential multi-call tool responses re-enabled** IMPROVED 30

Re-enables multi-call responses, executed sequentially in provider order.

– Brief statement with no example of use v0.1.2026.10.3

20 **Live gateway activity display in shell** IMPROVED 30

Shows live gateway activity in the shell and shortens the CI demo.

– States the feature without further mechanism v0.1.2026.10.3

21 **Default hosted LLM model set to GPT-6.1 Sol** IMPROVED 30

Hosted LLM turns now default to `GPT-6.1 Sol` at extra high.

– Only a one-line default change noted v0.1.2026.10.10

v0.1.2026.10.8 NOTES ↗ STABLE

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.2026.10.8 https://github.com/Tracer-Cloud/opensre.git
# already have the repo? check out this version:
$ git checkout v0.1.2026.10.8
```

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.155 NOTES ↗ RANK

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

> HyperFrames added configurable video quality and HDR output controls to its render endpoints, with stricter input validation across both render servers.

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

01 **Quality and HDR controls on render endpoints** NEW 90

The Producer render server's `/render` and `/render/stream` endpoints gain `crf`, `videoBitrate`, and `videoFrameFormat` fields, with invalid `videoFrameFormat` values now returning HTTP 400. The Studio Server render route gains the same `crf`, `videoBitrate`, and `videoFrameFormat` fields plus `hdrMode`; out-of-range values return HTTP 400, and supplying both `crf` and `videoBitrate` together is rejected with an error.

Control output quality on a Producer render job by specifying a CRF value — useful when you need consistent quality without a fixed bitrate ceiling.

```
$ curl -X POST http://localhost:3000/render -H 'Content-Type:
application/json' -d '{"crf": 23, "videoFrameFormat": "yuv420p"}'
```

Enable HDR output on a Studio Server render by supplying `hdrMode` alongside quality settings — pass only `crf` or `videoBitrate`, never both.

```
$ curl -X POST http://localhost:4000/render -H 'Content-Type:
application/json' -d '{"crf": 18, "videoFrameFormat":
"yuv420p10le", "hdrMode": true}'
```

— Names exact endpoints, fields, and validation behavior with runnable examples. v0.8.155

WAS THIS USEFUL? 👍 👎

Snowflake Cortex AI

Feedback ⓘ

Oct 13, 2026: Cortex Agent SDK (Public Preview) NOTES ↗ RANK

Snowflake Cortex AI provides managed language-model functions, search, and agents that work with data governed in Snowflake.

> Snowflake Cortex AI introduced the Cortex Agent SDK in Public Preview, giving developers Python and TypeScript libraries to build agentic applications on Cortex Agents and Cortex Code.

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

01 **Cortex Agent SDK for Python and TypeScript** NEW 45

The Cortex Agent SDK launches in Public Preview, letting applications run Cortex Agents and Cortex Code, maintain conversations and threads, observe durable turns, and reconnect to long-running work.

— Names SDK and languages but no code sample or API detail

Oct 13, 2026: Cortex Agent SDK (Public …

WAS THIS USEFUL? 👍 👎

Hugging Face OpenEnv

Feedback ⓘ v0.8.0 NOTES ↗ RANK

An interface library for RL post training with environments.

> OpenEnv's latest releases add a production `/harness` WebSocket route with producer-owned training captures for TRL via RFC 006 harness interception, rework the Harbor serve web UI for public deployment, and deprecate `opencode_env` / `pi_env` in favor of `harbor_env`, alongside new session telemetry and Level 2 validation probes.

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

01 **Production `/harness` WebSocket route** NEW 65

Adds a production `/harness` WebSocket route with mode wiring for agentic harness sessions.

– Names the exact endpoint but no further usage detail. v0.7.0

02 **Deprecation of `opencode_env` and `pi_env`** DEPRECATED 65

`opencode_env` and `pi_env` are deprecated in favour of `harbor_env` and will be removed in 0.8.0.

– Exact modules and removal version give a clear migration path. v0.7.0

03 **Training trace capture via RFC 006 harness interception** NEW 60

Adds producer-owned training captures for harness sessions, enabling token-faithful traces for TRL via RFC 006 harness interception.

– Names RFC and TRL integration but no example given. v0.7.0

THINNER COVERAGE BELOW

04 **`openenv.core.harness` package restructuring** IMPROVED 50

Splits `openenv.core.harness` into a package and introduces foundation types for agentic harnesses.

– Names the module path but not the new types. v0.7.0

05 **Level 2 validation probes for environments** NEW 35

Introduces Level 2 validation probes for deeper environment validation, adding validation of runtime repeatability and validation of discovery and rubric attribution.

— Names validation types but no API or flag to act on. v0.8.0

06 **Harbor serve web UI public deployment rework** IMPROVED 30

Reworks the Harbor serve web UI to make it safe to deploy publicly.

— States the outcome without describing the mechanism. v0.7.0

07 **Session telemetry for runtime tracking** NEW 20

Adds session telemetry to track runtime sessions.

— Bare description with no mechanism or surface named. v0.8.0

└──▷ **BREAKING ON UPGRADE**

! `opencode_env` and `pi_env` are deprecated in favour of `harbor_env` and will be removed in 0.8.0.

WAS THIS USEFUL?  

◆ Local LLM Runtimes

Comfy Org ComfyUI

Feedback ⓘ v0.40.0 NOTES ↗

ComfyUI is a node-graph interface, API, and backend for running diffusion models, chaining generation steps into reusable visual workflows.

Adds native LoRA stack loader nodes, a camera-info node system, asset export API, `--enable-assets` / `--disable-assets` flags, metadata on websocket messages, and EXR compression/previews.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.40.0 https://github.com/Comfy-Org/ComfyUI.git
# already have the repo? check out this version:
$ git checkout v0.40.0
```

Adds `--disable-assets` and `--enable-assets` CLI flags to control the asset scanning system; assets are now on by default for the Windows standalone build.

Adds native `LoRA stack loader` nodes (CORE-186) for stacking LoRAs without custom node dependencies.

Adds `LatentOperationBlend` node and `start` / `end` percent parameters to `LatentApplyOperation`.

Adds `CreateCameraInfo` node and `CameraInfoState` widget input for structured camera metadata in workflows.

Adds `CameraAngle` node with `camera_info` and `shot prompt` outputs.

+8 more

NEW INTEGRATION Google Nano Banana 2.1 partner node

NEW INTEGRATION Quiver V2 SVG nodes

NEW INTEGRATION Vidu Q4 model partner nodes

NEW INTEGRATION ClarityAI Crystal Upscaler partner node

NEW INTEGRATION Kling Virtual Try-On partner node

UPDATED INTEGRATION HeyGen Video 1.0 nodes: 2K resolution support

NEW INTEGRATION Higgsfield Genjutsu Motion Transfer partner node

NEW INTEGRATION FishAudio Drama 3 model partner node

NEW SUPPORTED PLATFORM ARM64 Nvidia portable build

WAS THIS USEFUL?  

vMLX

Feedback ⓘ v1.6.78 NOTES ↗ RANK

The vMLX server runs compressed MLX models on Apple Silicon with disk caching, paged memory, continuous batching, and hybrid SSM scheduling.

> vMLX 1.6.78 adds multimodal model support, new decoding backends, faster SSD cache restoration with preserved generation state, refined API statistics, and a macOS 26 Tahoe build.

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

01 **MiniCPM-V-4.7 multimodal support** NEW 65

Adds MiniCPM-V-4.7 text, image, and video support with bundled DSpark detection and exact verification enabled by default; bundles without a drafter use ordinary decoding.

– Names model and mechanism but no config/flag to invoke it v1.6.78

02 **Faster SSD cache restoration with state preservation** IMPROVED 65

Publishes completed responses before background SSD persistence finishes and synchronizes dependent continuations with their cache writes, reducing redundant snapshot copies and grouping compatible block files for faster restoration. Native recurrent and attention state is preserved across SSD restoration, media turns, and restart, reducing retained snapshot memory and shared-pool reconciliation overhead.

– Detailed mechanism but no command or config surface to act on v1.6.78

03 **macOS 26 Tahoe build support** NEW 65

Adds a macOS 26 Tahoe build alongside the existing macOS 14.5+ Sequoia build, both containing the same application and engine changes with platform-appropriate MLX wheels.

– Names exact OS versions and wheel targeting for install choice v1.6.78

THINNER COVERAGE BELOW

04 **API statistics and generation control improvements** IMPROVED 55

Improves cache, prefill, and tool-turn statistics and preserves native generation controls across the Chat Completions and Responses APIs.

– Names the two APIs affected as a starting point v1.6.78

05 **GLM DFlash2 and related execution integrations** NEW 35

Adds GLM DFlash2 integration alongside Naive N0.5 and GLM-5.3 Flash JANGHT execution improvements.

— Names integrations but gives no mechanism or usage detail v1.6.78

WAS THIS USEFUL? 👍 👎

Ollama

Feedback ⓘ v0.40.4 NOTES ↗ CODE ↗ RANK

Local large language model runner that downloads, serves, and chats with models through a CLI and REST API.

> Ollama v0.40.4 adds CLI support for decision-model queries, multimodal embeddings via llama-server, and tighter Codex integration including a live model picker and usage-credit support for `ollama launch`.

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

01 **Decision model queries via `--ask` flag** NEW 85

The new `--ask` flag on `ollama run` lets you pipe text into a decision model to ask yes/no questions, classify into labels, or score against a rubric directly from the terminal, e.g. `ollama run nimble --ask "Is a refund being requested?" < ticket.txt`.

Classify an incoming support ticket to detect refund requests without writing any code — pipe the ticket body straight into a decision model.

```
$ ollama run nimble --ask "Is a refund being requested?" <
ticket.txt
```

— Exact flag and runnable example given v0.40.4

THINNER COVERAGE BELOW

02 **Codex launch model picker and usage credits** IMPROVED 55

`ollama launch codex` now populates the `/model` picker with locally installed Ollama models, letting you swap models mid-session with `/model`. `ollama launch` also now works with usage credits without requiring an upgrade to a paid plan.

— Names commands and UI path but no full walkthrough v0.40.4

03 **Decision models on MLX for Apple silicon** NEW 25

Decision models (Kev scoring) are now supported by the MLX runner on Apple silicon.

— States support added but no usage detail v0.40.4

04 **Multimodal embedding support** NEW 20

Adds multimodal embedding support through llama-server.

— No mechanism, scope or example given v0.40.4

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Pinecone

Feedback ⓘ snapshot-20261014 NOTES ↗

Pinecone is a managed vector database that stores and queries embeddings for AI applications.

Pinecone adds BYOC deployment (AWS/GCP/Azure) via setup wizard and Pulumi, BYOC index creation via standard API, and a `/llms.txt` documentation index endpoint.

NEW SUPPORTED PLATFORM Adds BYOC deployment into AWS, GCP, or Azure accounts via a setup wizard and Pulumi, with public or private networking options.

Enables index creation inside a BYOC environment using the standard Pinecone API.

Exposes `/llms.txt` as a documentation index endpoint to discover all available pages programmatically.

WAS THIS USEFUL? 👍 👎

ralforion OrionBelt Ontology Builder

Feedback ⓘ

v1.29.1 NOTES ↗ RANK

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

> OrionBelt v1.29.1 adds live progress feedback to its long-running check panels so large ontologies no longer appear to freeze during validation.

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

01 Progress bars for Quality and SHACL checks IMPROVED

65

A progress bar now appears under the Run Quality Checks button showing the current rule as it executes (e.g. 'Rule 3 of 18: Duplicate preferred labels...'), and a progress bar appears under the Run SHACL check button showing a distinct model-to-shapes generation step followed by the shapes being validated, including a count when multiple sources are selected. Both bars clear automatically once results arrive.

Monitor rule-by-rule progress on a large ontology instead of waiting on a frozen screen.

📍 1. Open your ontology in OrionBelt. 2. Click Run Quality Checks. 3. Watch the progress bar beneath the button — it updates per rule, e.g. 'Rule 3 of 18: Duplicate preferred labels...' — and clears automatically when results arrive.

Track SHACL shape validation progress, including the shape-generation step, when checking against multiple sources.

📍 1. Tick one or more SHACL sources in the OrionBelt panel. 2. Click Run SHACL check. 3. The progress bar beneath the button first shows the shape-generation step, then cycles through shapes being checked (with a source count when more than one is selected), and clears when validation completes.

— Names exact buttons and progress text; UI-only path, no API. v1.29.1

WAS THIS USEFUL? 👍 👎

ai-memory

Feedback ⓘ v3.0.0 NOTES ↗ RANK

Solution for long term memory for agent coding CLIs and to facilitate handoff between different agent vendors

> ai-memory v3.0.0 introduces a typed cross-agent handoff protocol, support for more than twenty AI coding agent harnesses, zero-LLM fused memory search, and a first-party Claude Code plugin alongside new install tooling and distribution artifacts.

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

01 **Setup command and version-pinned install** NEW 75

Adds the `ai-memory setup` one-step command to initialize the tool after install, and the `AI_MEMORY_VERSION` environment variable to control which version is pulled during install via the `install.sh` script.

Bootstrap ai-memory on a new machine so all subsequent agent sessions share the same memory store.

```
$ curl -fsSL https://github.com/akitaonrails/ai-memory/releases/download/v3.0.0/install.sh | AI_MEMORY_VERSION=3.0.0 sh
ai-memory setup
```

— Exact command and env var given with runnable install snippet v3.0.0

02 **New distribution artifacts across platforms** NEW 75

Ships `ai-memory-claude-plugin.zip` as a first-party Claude Code plugin, `ai-memory-hooks.tar.gz` and `ai-memory-install-hooks` for lifecycle hook capture across supported agents, `ai-memory-wrapper`, `ai-memory-wrapper.cmd`, and `ai-memory-wrapper.ps1` wrapper scripts, and a self-contained binary with NixOS module support, distributed for Linux (x86_64, aarch64), macOS (x86_64, aarch64), Windows x86_64, and Docker (`akitaonrails/ai-memory:3.0.0`).

— Every artifact and platform named, no usage steps given v3.0.0

03 **Support for 20+ AI coding agent harnesses** NEW 60

Supports more than twenty AI coding agent harnesses including Claude Code, Codex, Cursor, Gemini CLI, OpenCode (V1/V2 auto-detected, with compatibility aliases `opencode2`, `opencode-v2`, and `open-code2`), Grok, Devin, Kimi, Kiro, GitHub Copilot CLI, VS Code Copilot, Zed, Muse Code, and others, integrated via MCP registration, lifecycle hooks, or both.

— Names every agent and alias but no setup steps v3.0.0

04 **Cross-project user profile injection** NEW 60

Provides a cross-project user profile, documented in `docs/cross-project-profile.md`, that injects personal defaults (package manager, test layout, architecture habits) into every new repository at session start.

— Doc path named as starting point, no command shown v3.0.0

05 **Fused zero-LLM memory search** NEW 60

Supports full-text, entity, link, and optional vector search fused into a single ranking for memory recall, with zero LLM calls required by default.

— Mechanism and modes named, no invocation shown v3.0.0

THINNER COVERAGE BELOW

06 **Typed cross-agent handoff protocol** NEW 55

Handoffs are modeled as a typed protocol where each handoff has an owner and is claimed exactly once, letting agents like Claude Code and Codex hand off mid-task state to one another without re-explanation.

— Mechanism described but no command or API shown v3.0.0

07 **Background LLM dream pass for cold notes** NEW 50

Opt-in background LLM `dream` pass rewrites cold note clusters into single coherent pages while idle, canceling on activity, with pre-merge versions preserved.

— Clear mechanism but no toggle or flag named v3.0.0

08 **Access-weighted memory retention** NEW 45

Memory accessed via page opens, searches, or link traversals decays slower than unused memory, so frequently used notes persist longer.

— Behaviour described but no config or limit given v3.0.0

09 **Multi-user auth and audit log** NEW 40

Includes multi-user auth, per-person attribution, and an audit log recording every mutation, available at no paid tier.

— Feature named but no config or endpoint given v3.0.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261014 NOTES ↗

RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's October 2026 snapshot heavily expands its MCP tool surface — adding run querying, prompt management, deployment/log lookup, and sandbox file/command tools — while also improving sandbox proxy routing, dataset APIs, usage billing units, and audit logging.

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

01 **Header-based routing in sandbox proxy rules** NEW 80

Sandbox proxy rules now accept `match_headers` entries (e.g., `'authorization: Bearer account-b'`), enabling multiple rules on one host to each swap a different placeholder credential and serve multiple accounts.

Route multiple accounts through one sandbox proxy host by matching on Authorization header values, so each rule swaps its own placeholder credential.

yaml

```yaml
match_headers:
  - 'authorization: Bearer account-b'
```

— Named config field with concrete example and mechanism snapshot-20261014

02 **Dataset example replacement endpoint fix** IMPROVED 80

Replacing a dataset's examples via `PUT /v1/platform/datasets/{dataset_id}/examples` now succeeds when the request updates existing examples.

Replace all examples in a dataset via the API when bulk-updating a golden set for an experiment.

```
$ curl -X PUT
'https://api.smith.langchain.com/v1/platform/datasets/{dataset_id
}/examples' \
  -H 'Authorization: Bearer <api_key>' \
  -H 'Content-Type: application/json' \
  -d '{"examples": [{"id": "<example_id>", "inputs": {"q": "What
is LangSmith?"}, "outputs": {"a": "An observability
platform."}}]}'
```

— Exact endpoint with runnable curl example `snapshot-20261014`

---

03 **`query_runs` MCP tool replaces `fetch_runs`** NEW 70

Adds `query_runs` to the LangSmith MCP, accepting the same parameters as client.runs.query() with typed selects for costs, token breakdowns, full inputs/outputs, and cursor pagination, returning token counts and USD costs by default; `fetch_runs` is deprecated in favor of `query_runs` but now also returns costs and token counts.

— Names tools and parameters but no runnable example given `snapshot-20261014`

---

04 **Sandbox file, command, and usage tools in MCP** NEW 65

Adds MCP capabilities to read, write, and search sandbox files, stream long-running sandbox commands, share sandbox ports behind LangSmith login, and report sandbox usage, organization usage, audit logs, usage limits, trace retention settings, and organization members.

— Enumerates capabilities but lacks exact tool/endpoint names `snapshot-20261014`

---

05 **Audit logging for Managed Tools calls** NEW 65

Audit log entries for calls through LangSmith Managed Tools now appear as `invoke_mcp_server` or `invoke_mcp_gateway`, including the server or gateway, the tool called, request duration, and user agent.

— Names event types and fields but no lookup example `snapshot-20261014`

---

06 **Prompt management tools in LangSmith MCP** NEW 60

Adds `get_prompt_by_name` to the LangSmith MCP, returning the full prompt template, along with new MCP tools for agents to push prompt versions, browse commits, manage commit tags, update or delete prompts, and read and commit Context Hub agent and skill files.

— Lists tool capabilities but no usage example `snapshot-20261014`

THINNER COVERAGE BELOW

07 **Deployment and log lookup tools in MCP** NEW 50

Adds MCP tools for agents to find deployments and revisions and read build or runtime logs, filtered by revision, time, search text, and severity.

— Describes filters but no concrete tool names or examples snapshot-20261014

08 **LangChain Standard Units for Engine spend** IMPROVED 50

Engine usage, spend limits, and pricing now display in LangChain Standard Units (LSUs), priced at $1 each.

— States pricing unit but no navigation or config detail snapshot-20261014

WAS THIS USEFUL?  

UK AI Security Institute Inspect AI

Feedback ⓘ

0.3.280 CODE ↗

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

Inspect AI 0.3.280 adds response-header capture on model calls, richer sentinel hooks for tool-stage monitors and whole-turn human approval, and agent-bridge filter improvements.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.3.280
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.280
```

Preserves the response headers of a model call on the model output, making rate-limit and tracing metadata available to evaluation code.

Enables agent-bridge filters to identify which agent a bridged model request belongs to.

Sentinel tool-stage monitors now receive the list of offered tools and a flag indicating whether the result was truncated.

Allows a sentinel to ask a human to approve or terminate a whole assistant turn.

NEW SUPPORTED PLATFORM Gemini 4 Argon support

└──▷ BREAKING ON UPGRADE

! Docker sandboxes now block writable host mounts by default; existing setups using writable host mounts will fail on upgrade. Switch to read-only mounts, copy inputs with `Sample.files`, use Docker-managed volumes, or use the documented unsafe opt-in. Read-only host mounts and Compose `configs` and `secrets` now require Docker Engine 25+, Compose 2.24+, and Linux kernel 5.12+ on the Docker host.

! The short-lived `inspect_ai.core` package introduced in 0.3.277 is removed; core data types revert to their original import paths.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261014 NOTES ↗ RANK

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

> Braintrust's window centers on Loop/Patterns investigation upgrades (severity, duplicate consolidation, trace evidence, PagerDuty alerting) alongside Anthropic span metadata improvements, Windows DPAPI credential storage, a new eval `cancel_event` parameter, and breaking field renames on `pi` model call spans.

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

01 Breaking renames on `pi` model call spans

BREAKING

95

On `pi` model call spans, `reasoning_tokens` is renamed to `completion_reasoning_tokens`, `cost` is renamed to `estimated_cost`, `metadata.provider_request` is replaced by top-level fields such as `temperature` and `tools`, and skill loads now use `tool_name: 'read'` with `tool_kind: 'skill'` — update any queries that reference the earlier field names. Spans no longer include `metadata.bt_daemon_version`.

— Exact field renames with explicit migration guidance snapshot-20261014

02 PagerDuty destinations for Loop and alerts

NEW

90

Adds PagerDuty notification destinations for Loop automations and for log, time window, and environment alerts, configured with an integration key and incident severity per destination (requires data plane v2.17.0+). In the Braintrust console, go to Automations, open or create a log/time-window/environment alert, add a PagerDuty destination, enter the integration key, and set the desired incident severity.

Route a Braintrust log or time-window automation alert directly to an on-call PagerDuty service when an anomaly is detected.

> 📍 In the Braintrust console, go to Automations, open or create a log/time-window/environment alert, add a PagerDuty destination, enter your integration key, and set the desired incident severity.

— Exact console navigation and config steps given snapshot-20261014

03 Cancel event for early eval termination

NEW

85

Supports stopping a long-running evaluation early via a `cancel_event` passed to `Eval`, letting the task check hooks.cancel_event.is_set() and return early so CI pipelines can cancel runs after a deadline.

Stop a long-running evaluation early from a CI pipeline by signalling the cancel event after a deadline.

```python
import threading
from braintrust import Eval

cancel = threading.Event()

async def my_task(input, hooks):
    if hooks.cancel_event.is_set():
        return None
    return do_work(input)

await Eval('my-project', data=cases, task=my_task,
cancel_event=cancel)
```

— Runnable code example demonstrating the parameter `snapshot-20261014`

04 **Anthropic span diagnostics and MCP tool spans** IMPROVED

80

Adds Anthropic `diagnostics` request parameter recording on spans, plus `cache_miss_reason` and `cache_missed_input_tokens` in metadata when Anthropic reports a prompt cache miss, and changes Anthropic MCP connector tool calls to appear as child tool spans named after the MCP tool with full tool input (including streamed responses), replacing generic `mcp` spans.

— Exact field names and before/after span behavior given `snapshot-20261014`

05 **New Patterns investigation features (public preview)** NEW

65

In Patterns (public preview), adds duplicate-finding consolidation on Braintrust cloud that merges duplicate findings into one canonical pattern while linking closed duplicates, trace evidence inspection beside a pattern, and Pattern severity on Findings (requires data plane v2.18.0+) with optional correction and reason preserved across Loop investigations.

— Merge mechanism described, no usage path Just a feature name, no mechanism given Version requirement and preserved fields named

`snapshot-20261014`

06 **DPAPI-encrypted Windows credential storage** NEW

60

Adds DPAPI-encrypted credential storage on Windows at `%USERPROFILE%\.braintrust\auth\credentials.dpapi`, shared across terminal and MSIX desktop sessions.

– Exact file path given though change is automatic `snapshot-20261014`

THINNER COVERAGE BELOW

07 **Trace view relabels Decision spans** IMPROVED 55

The trace view relabels TypeSafe spans as Decision spans and lists them under Decision in the Filter by span type control.

– Exact UI label and filter control named `product docs`

08 **Protected fields for log redaction** NEW 50

Adds protected fields that detect and replace sensitive text in incoming project logs, configurable for a single project or for all projects across an organization.

– Scope described but no exact config key or UI path `product docs`

09 **Patterns Frequency chart default change** IMPROVED 50

The Patterns Frequency chart (public preview) now defaults to showing the share of all traces in the project, and uses overall frequency for saved percentages and frequency ranking.

– Clear before/after default behavior described `product docs`

10 **Console UI productivity improvements** NEW 45

The Braintrust console adds a split view to the query editor so two queries can be run and compared side by side, translate and copy actions for LLM responses and user messages in the new thread view, and clipboard image paste support in the composer.

– Named UI areas but no exact navigation steps `product docs`

11 **Continue in Loop for fresh investigations** NEW 40

Adds Continue in Loop (public preview), letting users start a fresh chat when a conversation is already active, keeping pattern investigations isolated.

– Purpose explained, no usage steps given `product docs`

└──▷ **BREAKING ON UPGRADE**

! On `pi` model call spans, `reasoning_tokens` is renamed to `completion_reasoning_tokens`, `cost` is renamed to `estimated_cost`, `metadata.provider_request` is replaced by top-level fields such as `temperature` and `tools`, and skill loads now use `tool_name: 'read'` with `tool_kind: 'skill'` — update any queries that reference the earlier field names.

! Spans no longer include `metadata.bt_daemon_version`.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

NVIDIA OpenShell

Feedback ⓘ v0.1.4 NOTES ↗ RANK

OpenShell runs autonomous AI agents in policy-controlled sandboxes with isolated filesystem, network, and credential access.

> OpenShell v0.1.4 adds gRPC server reflection to its gateway, overhauls provider credential handling with operator-only retrieval and stable placeholders, and extends the TypeScript and Python SDKs with new sandbox metadata, log streaming, and status APIs.

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

01 gRPC server reflection in gateway NEW

88

The gateway now supports gRPC server reflection, enabling `grpcurl -plaintext localhost:18080 list` and `describe` to enumerate services and messages without supplying proto files, for example `grpcurl -plaintext localhost:18080 describe openshell.v1.OpenShell` or calling Health directly with `grpcurl -plaintext -d '{}' localhost:18080 openshell.v1.OpenShell/Health`.

Explore the gateway's gRPC API interactively — list services, inspect the schema, or call Health — without a local copy of the proto files.

```
$ grpcurl -plaintext localhost:18080 list
grpcurl -plaintext localhost:18080 describe
openshell.v1.OpenShell
grpcurl -plaintext -d '{}' localhost:18080
openshell.v1.OpenShell/Health
```

— Exact runnable grpcurl commands and port named v0.1.4

02 TypeScript SDK sandbox metadata and resumable log streaming

NEW

75

Adds `createdAtMs` field to `SandboxRef` in the TypeScript SDK, exposing sandbox creation time in milliseconds, and adds `watchLogs` with loss-aware resume, allowing log streaming to recover from gaps without restarting from the beginning.

— Named SDK field and method but no usage snippet v0.1.4

THINNER COVERAGE BELOW

03 Operator credential retrieval and stable placeholders in providers

NEW

55

Adds operator-only provider credential retrieval so gateway operators can fetch provider credentials without exposing them to sandbox workloads, and adds support for stable credential placeholders in providers, letting policies reference credentials by stable name across rotations.

— Mechanism described but no concrete command or config key v0.1.4

---

04 **Snap gateway runs as user service by default** BREAKING

40

The snap gateway now runs as a user service by default instead of a system service.

— Behavior change noted but no migration guidance given v0.1.4

---

05 **CLI label creation and filtering for providers** NEW

35

Adds CLI label creation and filtering for providers.

— Names the CLI capability but no command syntax given v0.1.4

---

06 **Sandbox status conditions in Python SDK** NEW

25

Exposes sandbox status conditions in the Python SDK.

— One-line mention with no detail on fields or usage v0.1.4

WAS THIS USEFUL? 👍 👎

ToolHive

Feedback ⓘ v0.53.0 NOTES ↗ RANK

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

> ToolHive v0.53.0 adds group-based upstream filtering to the vMCP embedded auth server and deprecates the Kubernetes operator's permission profile field in favor of native Kubernetes security controls.

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

01 Kubernetes operator drops permissionProfile field

BREAKING

81

The Kubernetes operator now ignores `MCPServer.spec.permissionProfile`; MCPServers that set this field may have their proxy pods rolled once after upgrading. Operators should use `podTemplateSpec` security contexts and `NetworkPolicy` resources instead, and `thv export --format k8s` no longer emits the field.

— Clear migration path and exact field/command names given v0.53.0

02 Group-based upstream filtering in vMCP auth server

NEW

78

The vMCP embedded auth server now supports group-based upstream filter configuration via `authserver-config.yaml`, routing users to upstream OAuth providers based on group claims returned by the first upstream. Users without an account on a given service skip that service's OAuth leg entirely, and the feature fails closed when the group claim is absent.

— Names config file and mechanism but no example usage v0.53.0

└──▷ **BREAKING ON UPGRADE**

! The Kubernetes operator now ignores `MCPServer.spec.permissionProfile`; MCPServers that set this field may have their proxy pods rolled once after upgrading. Use `podTemplateSpec` security contexts and `NetworkPolicy` resources instead. `thv export --format k8s` no longer emits the field.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

Meta Llama 3.3 70B Instruct Feedback ⓘ

price increase · Oct 13, 2026 MODEL ↗

Llama 3.3 70B Instruct is a chat model from Meta.

Price increase on OpenRouter (meta-llama/llama-3.3-70b-instruct): $0.22 → $0.59 in, $0.50 → $0.79 out per million tokens (+168%), first seen 13 Oct 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M2.7 Feedback ⓘ price increase · Oct 13, 2026 MODEL ↗

MiniMax M2.7 is a reasoning model from MiniMax.

Price increase on OpenRouter (minimax/minimax-m2.7): $0.21 → $0.30 in, $0.84 → $1.20 out per million tokens (+43%), first seen 13 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-35B-A3B Feedback ⓘ

price increase · Oct 13, 2026 MODEL ↗

Qwen3.5-35B-A3B is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3.5-35b-a3b): $0.08 → $0.15 in, $0.75 → $1.00 out per million tokens (+87%), first seen 13 Oct 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 14, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue