THE AI TOOLCHAIN

NO. 070

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 22, 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 403 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 75 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. Opik now auto-scores live production traces with judge, decision-model and Python evaluators, so quality ssions surface without a manual sampling pass. Anthropic added

vim ? help

sandboxed bash and text-editor execution plus workspace-scoped keys and rate limits. And Cloudflare Workers AI's single `createAI` provider puts Workers models, gateway catalog, vendor models and fallback behind one call.

EVALUATE

Score production traces as they arrive instead of sampling them by hand later

Opik's online evaluation rules attach LLM-as-a-Judge, decision-model or Python evaluators directly to live traces, threads and spans, so the offline batch job that someone has to remember to run goes away; Phoenix moves its classification evaluators onto AI SDK decision models via `experimental_decide`. The practical shift is that a quality drop shows up in the trace view rather than in next week's eval run.

Opik · Arize Phoenix

DEPLOY

Route every model call through one provider object with fallback already wired in

Workers AI's unified `createAI` merges Workers AI models, AI Gateway catalog models, dynamic routes and vendor models with fallback, BYOK and stream-resume, so swapping or failing over a model is a config change rather than a second SDK path. Ollama v0.40.8 adds Flint, Recursive's 40B mixture-of-experts coding model, runnable locally from the CLI.

Cloudflare Workers AI · Ollama

GOVERN

Cap what an agent can reach and spend before it is loose in production

Anthropic's workspace-scoped rate limits, keys and reporting mean one team's runaway agent loop no longer eats another's quota, and org-wide web fetch/search domain controls are enforced inside Claude Managed Agents rather than left to prompt discipline. Otari ships organization-owned guardrail authoring and enforcement with end-to-end session tracing, so the policy and the evidence that it held live in the same place.

Anthropic · Mozilla AI Otari

BUILD

◆ Model Training & Fine-Tuning

PyTorch

Feedback ⓘ announced · Oct 22, 2026 BLOG ↗

Tensors and Dynamic neural networks in Python with strong GPU acceleration

TorchNeuron is a new open-source Apache 2.0 native PyTorch backend for AWS Trainium, adding the `'neuron'` device, adaptive eager mode, `torch.compile` with `backend='neuron'`, NKI custom kernel registration via `nki_op`, distributed training APIs, streams/events, and integrated profiling via `torch.profiler` with `NeuronConfig`.

Adds `'neuron'` as a native PyTorch device target via .to('neuron'), letting models run on AWS Trainium with no other code changes.

NEW INTEGRATION Adds torch.compiler.set_default_backend('neuron') and torch.compile(model, backend='neuron') to compile models for Trainium via TorchDynamo → AOTAutograd → Torch-MLIR → StableHLO → NEFF.

Introduces adaptive eager mode (experimental) on Trainium, fusing operations to approach `torch.compile` performance without a code change while keeping the interactive eager experience.

Adds `torch_neuronx.nki_op` decorator to register custom NKI kernels as standard PyTorch operators with `register_autograd` support for forward and backward passes.

Adds `torch_neuronx.Stream` and `torch_neuronx.Event`, mapping PyTorch stream semantics onto Trainium hardware queues, with the device-agnostic `torch.Stream`, `torch.Event`, and `torch.accelerator` entry points reaching the same implementation.

+6 more

NEW SUPPORTED PLATFORM AWS Trainium2 and Trainium3 support in PyTorch via TorchNeuron

NEW INTEGRATION TorchTitan integrated into veRL as a training backend

NEW INTEGRATION TorchTitan integrated into Miles

NEW INTEGRATION TorchTitan is the default training backend for Meshy

NEW SUPPORTED PLATFORM PyTorch-native training support for Muse Spark Open, Muse Glimmer, Qwen3-VL, Qwen3.5, Kimi K2.7, Kimi K3, and DeepSeek-V4

NEW SUPPORTED PLATFORM Dist-MoE support for NVIDIA SM100+ (Blackwell) GPUs

▾ 1 more release in this issue · seen 2026-10-22

> announced · Oct 22, 2026 BLOG ↗

TorchTitan adds TitanRL (RL training), FlexShard, GraphTrainer, torch_checkpointing, a new NCCL backend, local_compile, and torch_remat.

Introduces TitanRL, a reinforcement-learning training path sharing one model across pre-training, post-training, and vLLM inference, with asynchronous rollouts and agentic workload support.

Adds TitanRL batch-invariant mode producing bitwise-identical log probabilities between trainer and generator for precise RL numerical debugging.

Integrates `FlexShard`, a bucket-centric data-parallelism library with `Owned` and `BlockShard` placements enabling distributed Muon optimizer and blockwise-quantized FP8 AllGathers.

Adds `GraphTrainer`, a graph-mode training path with cross-module fusions, explicit graph-level debugging, and heterogeneous accelerator support.

Integrates a new opt-in NCCL backend (from `torchcomms` into `c10d`) with one-sided RMA windows, in-place process-group reconfiguration, eager communicator initialization, and raw NCCL shrink/grow/suspend/resume APIs.

+4 more

WAS THIS USEFUL?  

◆ AI Coding Agents

Vercel v0

Feedback ⓘ snapshot-20261022 NOTES ↗ RANK

The v0 generative UI platform builds web applications from natural-language prompts.

> v0 adds a Slack MCP integration for pulling chat context into prompts, and lets ChatGPT Plus/Pro subscribers bill AI responses to their existing ChatGPT plan instead of v0 credits.

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

01 **ChatGPT plan billing for AI responses** NEW 95

Users can cover eligible AI responses with a ChatGPT Plus or Pro subscription instead of v0 credits, for supported models `GPT-6 Astra`, `GPT-5.6 Sol`, and `GPT-5.6 Luna`; it's connected via the credit balance menu under `Use ChatGPT plan`. When active, the header shows a `ChatGPT` indicator and the model picker shows `Using ChatGPT plan`; if the ChatGPT usage limit is reached, v0 pauses generation, saves progress, and offers `Continue with v0 credits` rather than silently switching billing.

Route AI response costs to your existing ChatGPT subscription to avoid spending v0 credits on GPT-model generations.

📍 In v0, open the credit balance menu in the header → select `Use ChatGPT plan` → `Continue with ChatGPT` → approve subscription sharing. Then pick `GPT-5.6 Sol` at Standard speed and build normally.

— Names models, UI controls, and fallback behavior precisely snapshot-20261022

02 **Slack MCP integration in chat composer** NEW 75

v0 adds a Slack MCP integration reachable via `+` → `MCPs` in the chat composer, letting v0 pull Slack messages as context or post updates to a connected workspace.

Connect your Slack workspace so v0 can pull relevant thread context into a code change request without copy-pasting.

📍 In the v0 chat composer, click `+` → `MCPs`, choose `Slack`, and connect your workspace. Then prompt: 'Use the #frontend Slack discussion about the login redesign as context for updating the auth component.'

— Names exact UI path and mechanism, lacks posting details snapshot-20261022

WAS THIS USEFUL? 👍 👎

Yuval Sarel cones

Feedback ⓘ v0.4.4 NOTES ↗ RANK

Cones is a terminal dashboard for managing coding-agent sessions across projects.

> cones v0.4.4 reorganizes how launched sessions appear in the dashboard and consolidates fork/launch state into a single file.

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

01 **Session nesting under parent agents in launch view** IMPROVED 80

Sessions started with `cones launch` are now listed beneath their parent agent, marked `↳ ⚙`, with dimmed folder context shown when the launch runs in a different directory. Setting `launched_under_parent: false` in config keeps each launch listed in its own folder rather than nested under its parent agent.

— Names exact marker, command and config key with value v0.4.4

02 **Unified fork/launch state storage** IMPROVED 65

Forks and launches now share a unified `STATE_DIR/relations.json` state file; existing `forks.json` files continue to be read until cones writes the new file.

— Names file paths but no direct user action v0.4.4

WAS THIS USEFUL? 👍 👎

Token Optimizer

Feedback ⓘ v5.13.57 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's v5.13.57 release adds diagnostic visibility into Codex hooks and fixes a Windows-specific hook execution bug.

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

01 **codex-doctor hook diagnostics** NEW 60

`codex-doctor` now recognizes currently configured hooks, including when the Stop hook last ran and what it did.

— Names the command and what it reports, no deeper mechanism given v5.13.57

THINNER COVERAGE BELOW

02 **Codex hook execution fixed on Windows** IMPROVED 20

Fixes an issue where Codex hook execution was broken on Windows.

— Only mentioned in summary, no detail on cause or scope v5.13.57

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.163.0-alpha.11.2

NOTES ↗ CODE ↗ RANK

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

> OpenAI Codex CLI's biggest changes this window were TUI preset prompts with inline model/permission switching and a new `--cloud-skills-snapshot` flag for the app-server, alongside remote exec-server credential masking, a file-copy tool for remote environments, and Windows-specific sandbox and requirements-path hardening.

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

01 **Windows registry override for requirements path** NEW 90

Codex now reads `RequirementsPath` from `HKLM\SOFTWARE\Policies\OpenAI\Codex` (64-bit registry view) to override the system requirements file location on Windows, falling back to the default `OpenAI\Codex\requirements.toml` under ProgramData when the key is absent.

Point Codex at a custom enterprise-managed requirements file on a Windows fleet by setting `RequirementsPath` in the registry — no per-user file needed.

```
$ reg add "HKLM\SOFTWARE\Policies\OpenAI\Codex" /v
RequirementsPath /t REG_SZ /d
"C:\ProgramData\OpenAI\Codex\custom-requirements.toml" /f
```

— Named registry key with runnable example command rust-v0.163.0-alpha.11.2

02 **Cloud skills snapshot flag for app-server** NEW 80

Adds `--cloud-skills-snapshot <PATH>` flag to the app-server for loading frozen cloud skills from a JSON snapshot instead of a live skill service, useful for offline or reproducible environments.

Load a frozen cloud-skills snapshot when running the app-server offline or in a reproducible test environment, bypassing the live skill service.

```
$ codex app-server --cloud-skills-snapshot /path/to/skills-
snapshot.json
```

Load a pre-captured cloud skills snapshot offline instead of hitting the live skill service — useful in air-gapped or reproducible CI environments.

```
$ codex-app-server --cloud-skills-snapshot ./skills-snapshot.json
```

— Named flag with two runnable example commands `rust-v0.164.0-alpha.11`

---

03 **MCP authentication and tool-call handling improvements**

IMPROVED 70

Improves MCP handling by preserving ChatGPT authentication for MCP servers at `https://gov.chatgpt.com`, allowing dynamic tool calls to execute in parallel, honoring MCP tool-list changes within a turn, and preferring resource-specific scopes during MCP OAuth discovery.

— Mechanism and named server detailed but no user-facing command `rust-v0.164.0-alpha.11`

---

04 **Preset prompts and slash commands via /presets** NEW 70

Adds `/presets` slash command for defining and reusing prompts and slash commands in the TUI, a keyboard shortcut menu for quick access to presets, and configuration support for TUI input presets.

— Named command and config support, no example shown `rust-v0.164.0-alpha.11`

---

05 **Inline model and permission switching in TUI** NEW 70

Supports inline model, reasoning, and speed selection directly within the `/model` slash command, and inline permission profile selection with the `/permissions` slash command, including saving current permission settings as presets.

— Two named slash commands with clear behaviour, no example `rust-v0.164.0-alpha.11`

---

06 **Windows sandbox migration and network root filtering** IMPROVED 65

Migrates the Windows MXC sandbox to `mxc-sdk`, replacing separate MXC crates and adapting request construction to its policy and optional environment fields; also filters uninspectable network roots from implicit `:root` read grants on Windows, retaining only inspectable shares and roots containing the policy or command working directory.

— Internal mechanism detailed but not user-actionable `rust-v0.163.0-alpha.11.2`

THINNER COVERAGE BELOW

---

07 **Credential masking for remote exec-server commands** NEW 55

Adds scoped executor credential auth and credential masking for remote exec-server commands, including Zsh shell snapshots and rotating stdin credentials; a later release confirms remote credential masking with Zsh shell snapshots.

— Mechanism named but no config key or command shown `rust-v0.164.0-alpha.11` `rust-v0.164.0-alpha.10`

---

08 **Session-scoped title and status-line presets** NEW 55

Adds session-only arguments to `/title` and `/statusline` for quick terminal title and status-line layout configuration, along with quick-dial presets for terminal title and status-line layouts.

— Named commands, behaviour clear, no example given `rust-v0.164.0-alpha.11`

---

09 **Image previews and Markdown syntax theming in TUI** NEW 45

Shows image previews in the fullscreen transcript view and applies syntax theme colors to Markdown prose in the TUI.

— Two thin UI additions with no navigation detail `rust-v0.164.0-alpha.10`

---

10 **Automatic model catalog reload on account change** IMPROVED 40

Reloads the model catalog automatically after account sign-in or sign-out without requiring a restart.

— Clear behaviour but no surface or command named `rust-v0.164.0-alpha.11`

---

11 **File-copy tool for remote environments** NEW 30

Adds an opt-in tool for copying files between environments.

— Bare description, no name, flag, or mechanism given `rust-v0.164.0-alpha.11`

WAS THIS USEFUL? 👍 👎

All Hands AI OpenHands

Feedback ⓘ v1.30.0 NOTES ↗

RANK

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

> OpenHands v1.30.0 adds visibility into built-in integration connection status within the automations setup flow.

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

01 Integration status shown in automations setup modal NEW

40

The automations setup choice modal now shows the connection state of built-in integrations directly, letting users see at a glance whether an integration is connected before proceeding.

— Describes the UI change but no mechanism or exact navigation path. v1.30.0

NEW INTEGRATION Recursive Flint model support

> announced · Oct 21, 2026 BLOG ↗
>
> OpenHands adds Recursive Flint as a supported model backend, available via OpenRouter or local inference on Apple Silicon and NVIDIA hardware.
>
> Supports Recursive Flint as a model backend via OpenRouter by configuring an OpenRouter API key and selecting the Flint model identifier.
>
> Supports running Recursive Flint locally by serving the Flint checkpoint and configuring OpenHands to use the local inference endpoint.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ sdk-typescript-v0.1.22

NOTES ↗ RANK

An open-source AI coding agent that lives in your terminal.

> Qwen Code shipped a major managed-agent and daemon lifecycle overhaul — `qwen daemon` subcommands, a `--check` preflight validator with PASS/WARN/FAIL/SKIP output, and persisted workspace artifact tracking — alongside TypeScript SDK session resume and multi-modal input, a Generated UI extension, numerous Web Shell additions, and two breaking changes: renamed `enable*` settings keys and removal of the `read_many_files` tool.

└──▷ **WHAT SHIPPED · 19 FEATURES** most completely described first ? what's the number?

01 **Local daemon lifecycle control via `serve`** NEW 85

Adds local daemon discovery and control through the `serve` subsystem, with `qwen daemon start`, `qwen daemon start --wait`, `qwen daemon restart`, and daemon status-detail subcommands for lifecycle control, plus authorization of daemon ancestor workflows for multi-level workflow chains.

Start the local daemon and block until it is ready to accept connections, useful in scripts or CI that need the daemon up before proceeding.

```
$ qwen daemon start --wait
```

— Names runnable daemon subcommands with a shell example v0.26.0 desktop-v0.26.0

02 **New config keys for loop protection, environment and memory behavior**

NEW 83

Adds a `model.loopProtection` config key that switches default loop-protection behavior from stopping a runaway turn to reminding the model instead, reducing false terminations in long agentic runs, along with the `settings.env` field for configuring environment variables directly in the settings file, the `contextWindowSize` configuration key, and the `memory.enableManagedAutoMemory` setting controlling whether managed auto-memory is active; hosts that disable it no longer see remember/dream requests admitted or managed-memory instructions in the system prompt.

Switch loop protection from hard-stopping a runaway turn to reminding the model, reducing false terminations in long agentic runs.

```yaml
model:
  loopProtection: remind
```

— Named config key with a runnable YAML example Three named config keys with concrete behavioural effect

v0.26.0 sdk-typescript-v0.1.22

---

03 **Preflight checks for managed-agent deployments** NEW

80

Adds a `--check` preflight flag to `managed-agent` for read-only environment validation before a session starts; it prints PASS, WARN, FAIL, or SKIP per check and exits nonzero on failure.

— Explains mechanism and exit behaviour but no sample invocation v0.26.0

desktop-v0.26.0

---

04 **Session resume support in TypeScript SDK** NEW 65

The TypeScript SDK adds `resume` and `continue` options and extended `authType` support, plus the ability to resume sessions by session ID via the `query` command.

— Named SDK options and command but no code sample sdk-typescript-v0.1.22

---

05 **Managed-agent workspace, artifact and session management** NEW

60

Managed-agent sessions gain local and SSH workspace file references, a persisted workspace artifact library via `qwen serve`, and an artifact-tracking subsystem recording files the agent creates or edits. Sessions can run concurrently on shared workspace storage, approvals can change during an active turn, and enabled skills can be imported at session creation. The managed-agent reference distribution ships a configuration template and systemd units for the single-host managed-agent topology and bootstraps shared deployment credentials, the attachment input budget was raised to 8 MiB, and the daemon gained session-targeted Skill and Extension refresh plus session-create warm-up for workspace-isolated bound sessions in the runtime broker.

— Enumerates many named additions but no single actionable step v0.26.0

desktop-v0.26.0

---

06 **Web Shell session and preview additions** NEW 60

The web shell adds per-model Time-to-First-Token (TTFT) and output speed in the Token Usage panel, listing and revoking paired devices, trajectory follow and pause controls, deleting selected message groups from session history, `PDF` preview using the native browser viewer, `DOCX` file preview across file entry points, a session MCP

catalog for managed-agent sessions, managed session rename support, and background shell task runtime details.

— Lists named UI features but gives only a navigation path `v0.26.0` `desktop-v0.26.0`

07 **Generated UI extension with model-rendered cards** NEW

60

Adds bounded generated HTML and JavaScript views for extensions, including an experimental Generated UI extension that renders model-generated HTML/CSS cards with bounded local JavaScript, letting users adjust inputs without triggering another model turn.

— Explains mechanism but the gating flag isn't named `v0.26.0` `desktop-v0.26.0`

THINNER COVERAGE BELOW

08 **Settings keys renamed from disable* to enable*** BREAKING

55

Settings keys previously named `disable*` are renamed to `enable*` (e.g. `disableX` becomes `enableX`); existing configs using the old negative naming will break.

— Clear migration note but no full key list `sdk-typescript-v0.1.22`

09 **read_many_files tool removed** BREAKING

55

The `read_many_files` tool is removed; its functionality moves to a `readManyFiles` utility restricted to user `@-commands` only.

— Names old and new surfaces but no migration steps `sdk-typescript-v0.1.22`

10 **Session history export command** NEW

50

Adds an export command for session history in Markdown and HTML formats.

— Names output formats but not the exact command `sdk-typescript-v0.1.22`

11 **Multi-modal input across content generators** NEW

45

Adds multi-modal input support (image, PDF, audio) across all content generators.

— Names supported media types but no usage path `sdk-typescript-v0.1.22`

12 **FORK_MODE for ProcessTransport Electron IPC** NEW

45

Adds `FORK_MODE` support to ProcessTransport for Electron IPC integration.

— Named mode and integration with no further detail `sdk-typescript-v0.1.22`

13 **WeCom chat-channel integration** NEW

40

Adds WeCom application callback support as a new chat-channel integration.

– Names the integration but no setup detail v0.26.0

14 **Real PTY backend for Bun runtime** NEW 40

Adds a real PTY backend for the Bun runtime.

– Bare statement with no further mechanism v0.26.0

15 **extra_body passthrough for OpenAI-compatible providers** NEW 35

Adds `extra_body` support for OpenAI-compatible providers.

– Named parameter with no usage context sdk-typescript-v0.1.22

16 **Concurrent runner for batch CLI execution** NEW 30

Adds a concurrent runner for batch CLI execution.

– Bare feature name without mechanism sdk-typescript-v0.1.22

17 **Zed extension for agent server** NEW 30

Adds a Zed extension for the Qwen Code agent server.

– Names the integration but no install or usage detail sdk-typescript-v0.1.22

18 **Reasoning effort on LLM telemetry spans** NEW 25

Records reasoning effort on LLM telemetry spans.

– One-line telemetry note with no further context desktop-v0.26.0

19 **Experimental LSP support** NEW 25

Adds experimental LSP support for code intelligence.

– Named capability only, no scope given sdk-typescript-v0.1.22

└──▷ **BREAKING ON UPGRADE**

! Settings keys previously named `disable*` are renamed to `enable*` (e.g. `disableX` becomes `enableX`); existing configs using the old negative naming will break.

! The `read_many_files` tool is removed; its functionality moves to a `readManyFiles` utility for user `@ -commands` only.

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.16.0 NOTES ↗

RANK

Zeroshot orchestrates executor-verifier agent loops that implement and validate software changes while logging workflow state to SQLite.

> Zeroshot v10.16.0 introduces human-managed machine delegations with expiring, rotatable API keys managed through the CLI.

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

01 **Machine delegations with expiring API keys** NEW 65

Adds human-managed machine delegations owned by a user or organization, with expiring API keys and overlapping rotation through the CLI. Keys inherit their owner's current authority and stay out of run inputs and subprocesses.

– Describes mechanism and scope but no exact CLI command names v10.16.0

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ Self-Hosted Orbs (2026-10-22)

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 shipped a native macOS/iOS app with voice-call support, a deep Linear integration for agent-driven issue work plus Slack mentions and generic MCP connections, expanded runner networking/browser sharing and GPU rendering, and refined thread snooze controls with CLI thread attachment.

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

01 **Network and browser sharing for orb runners** NEW 95

Adds `--share-network` and `--share-browser` flags to `amp --no-tui` so a local runner shares the host's network connection and signed-in Chrome cookies with orbs, bypassing cloud IP blocks. Adds `amp net route <runner-id>`, `amp orb desktop ensure`, and `amp sync browser` commands so agents can route traffic and sync Chrome cookies inside an orb automatically, and AGENTS.md (or Global AGENTS.md) can instruct every orb to set up this routing and browser sync before using the web.

Start a local runner that shares your network and Chrome cookies with orbs so they can access sites that block cloud IPs or require authentication.

```
$ amp --no-tui --runner-id home-mac --share-network --share-browser
```

Add to AGENTS.md so every orb automatically routes through your local runner and syncs your Chrome cookies before browsing.

```markdown
Before using the web in an orb, run
`amp net route home-mac`
`amp orb desktop ensure`
`amp sync browser`
```

— Named flags, commands and a runnable example command product docs

02 **Native Linear integration with agent routing** NEW 95

Workspace admins install Amp in Linear under Workspace Settings → Integrations & MCP, and members connect via Settings → Integrations & MCP or Connect Linear (installs Linear's MCP server) so orbs can search, read, create, and update issues and post

comments as that person. Assigning or mentioning `@Amp` in a Linear issue or comment starts a Linear agent session and orb thread running as the assigning user. Each installation's Agent Routing page adds a Default Project and per-team or per-project routes controlling the Amp project, mode, and orb provider per session, plus Start Instructions (up to 4,000 characters) giving every Linear-triggered session a standing workflow that per-route instructions can override. Event subscriptions let any member have Amp or Puck start a new orb for each new issue created in a named Linear team, with configurable agent mode, Amp project, optional label filter, expiration time, and optional follow-up on subsequent comments; orbs can also post comments as the installed Amp app rather than as the user.

— Extensive named settings paths and config options, UI-based product docs

03 **Snooze mode for automation and chat threads** NEW 75

Automation threads can run silently in snooze mode, letting scheduled runs and event subscription deliveries (GitHub CI results, Slack listener messages) continue in the background without hitting the sidebar or sending notifications; prompts can instruct the agent to unsnooze when a condition is met. Regular threads gain explicit snooze options — until next reply, until 8am, until Monday morning, or indefinitely — muting notifications while the agent keeps running; snoozed threads collect under a sidebar section and resurface automatically when the snooze ends, when approval is required, or when all automations complete, fail, or are cleared.

— Clear mechanism and options but only UI navigation, no command product docs

04 **GPU-accelerated runner desktop rendering** NEW 65

Supports the `AMP_DESKTOP_WLR_RENDERER` environment variable to enable GPU rendering (e.g. `gles2` or `vulkan`) for the runner desktop instead of the software-render default.

Try GPU-accelerated desktop rendering on a runner with a GPU render node available, instead of the default software renderer.

```
$ AMP_DESKTOP_WLR_RENDERER=gles2 amp --no-tui --desktop
```

— Env var and example command given, limited further detail product docs

05 **CLI attachment to orb threads via amp sync** NEW 65

Adds the `amp sync` command (with thread ID pre-filled) for attaching the local CLI to an orb thread, copied via Orb > Copy Sync CLI Command from the More Actions menu.

Attach your local CLI to an already-running orb thread so you can interact with it from the terminal.

```
$ amp sync <thread-id>
```

— Runnable command with clear UI source, thin otherwise product docs

06 **AMP_ORB environment variable for orb detection** NEW

65

Adds `AMP_ORB=1`, automatically set by Amp for all processes running inside an orb (setup scripts, services, and agent-run commands), so scripts and apps can branch behavior based on whether they are in an orb or on a local machine.

Gate orb-specific setup logic in a shell script so the same script works both locally and inside an Amp orb.

```
$ if [ "${AMP_ORB}" = "1" ]; then
  echo 'Running in an orb'
  # orb-specific initialisation
fi
```

— Named env var with a runnable shell example product docs

THINNER COVERAGE BELOW

07 **Native macOS and iOS app with voice calls** NEW 55

Adds a native macOS app (Apple silicon, macOS 26 or later) with voice calls running inside the Amp window, and iPhone support where calls run as native phone calls appearing in the status bar and lock screen with mute, speaker, headset, and hang-up controls available while the app is backgrounded.

— Describes behaviour and platform scope but no setup steps product docs

08 **Slack mentions and generic MCP integrations** NEW 40

Mentioning `@Amp` in Slack lets users ask questions, start work, or check on a thread, and any MCP-compatible service (such as Sentry or Notion) can be connected via the MCP configuration path.

— Named integrations but described only in one line each product docs

NEW INTEGRATION Self-hosted orb provider integrations

NEW SUPPORTED PLATFORM Kubernetes self-hosted orbs on GKE, EKS, AKS

> Self-Hosted Orbs (2026-10-22) BLOG ↗ LATEST STABLE

Amp adds `amp orb provider` and `amp orb bootstrap` commands plus `AMP_ORB_PROVIDER_TOKEN` to bring self-hosted orbs to any user-supplied compute infrastructure.

Adds `amp orb provider <dispatch-program>` command to run a self-hosted orb provider that dispatches `spawn`, `pause`, and `delete` operations to a user-supplied script or program.

Supports `AMP_ORB_PROVIDER_TOKEN` environment variable to authenticate the orb provider against the Amp workspace.

Dispatches orb lifecycle operations via `ORB_OPERATION`, `ORB_ID`, `ORB_IMAGE`, `ORB_TOKEN`, and `AMP_URL` environment variables passed to the dispatch program.

Adds `amp orb bootstrap` command to connect a spawned orb container back to Amp so the agent can be set up and run.

Provides a Helm chart with provider-specific setup guides (`gke.md`, `eks.md`, `aks.md`) and Terraform examples for running self-hosted orbs on GKE, EKS, or AKS.

WAS THIS USEFUL? 👍 👎

Earendil Works Pi

Feedback ⓘ v1.4.0 NOTES ↗ RANK

Pi is an AI coding agent toolkit that provides a unified LLM API, agent loop, TUI interface and CLI.

> Pi adds a new classifier model type with image input support across Jev, Clef and GPT-6 Luna, introduces config keys to control how that image input is sent, and ships several breaking changes to default models, extension headers and summary generation.

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

01 Image input configuration for classifier models NEW

90

Adds `imageInput` to `.pi` model configuration, accepting `"assets"`, `"images"`, or `"state-image-urls"` to control how classifier images are transmitted, auto-detected from provider, `baseUrl`, and model ID. Adds `sendImageOptInHeader` to control whether requests with images include the `TypeSafe-Danger-Zone-Images: true` header required by TypeSafe's experimental image input API, defaulting to `true` for `typesafe-system-one` unless OpenRouter is detected. Introduces `compat.imageInput` as the per-model config field controlling image input request format for System One services, set automatically for built-in models and configurable in a custom provider's `compat` block.

Override image transmission mode to `images` (Clef on Workers AI) when auto-detection picks the wrong format.

```yaml
imageInput: images
```

Disable the TypeSafe opt-in header when routing through OpenRouter to avoid unexpected header behaviour.

```yaml
sendImageOptInHeader: false
```

Enable image input for a proxy that serves Clef-compatible classifiers, using the new compat fields.

```yaml
imageInput: true
sendImageOptInHeader: true
```

– Exact config keys, values and defaults with runnable examples product docs

02 **Classifier models with image input support** NEW INTEGRATION

UPDATED INTEGRATION NEW

85

Pi adds classifier models that answer typed questions about JSON state — picking from choices, yes/no, or a scored probability — rather than chatting, supported via `typesafe/jev-1.14`, `typesafe/jev-1.14.0` (TypeSafe Jev), `cloudflare/clef`, `cloudflare/clef-flash`, `cloudflare/clef-omni`, `@cf/cloudflare/clef-omni` (Cloudflare Clef via Workers AI and OpenRouter), and OpenAI GPT-6 Luna. Classifier models enforce a hard limit of 922K input tokens, with requests over roughly 600K tokens subject to a ~5 second gateway timeout. Image input is available on models whose `includes` declares it: GPT-6 Luna, Jev (PNG/JPEG, experimental on TypeSafe's own API, text-only via TypeSafe's own endpoint), and Clef (PNG/JPEG/WebP, max 4 images per request), plus any OpenRouter model whose catalog lists image support. Provider-level `compat` support for classifier models now extends to extension classifier models too.

– Names exact model IDs, limits and per-model image formats product docs

THINNER COVERAGE BELOW

03 **Default models changed for Copilot and Xiaomi providers**

55

BREAKING

The default GitHub Copilot model changed from `gpt-5.4` (retired by Copilot) to a new default. The default Xiaomi model changed to `mimo-v2.6-pro` for the `xiaomi` and Xiaomi Token Plan providers, with `mimo-v2.5-pro` removed from the catalog. Users who pinned either provider should check their config.

– Names old and new model IDs but new Copilot default unnamed product docs

04 **Extension provider header visibility narrowed** BREAKING

50

Extensions' `before_provider_headers` handlers no longer see the OpenRouter, NVIDIA NIM, and OpenCode attribution headers; returning `null` for one still removes it.

– Names affected headers but no migration steps product docs

05 **Summary generation ignores output-cap parameter** BREAKING

50

generateSummary() and generateSummaryWithUsage() now ignore their output-cap parameter; summary requests use the model's output limit instead.

— Names exact functions and behaviour change product docs

06 **ChatGPT Business usage-limit error handling** IMPROVED

30

ChatGPT Business workspace usage-limit errors are no longer retried and now surface a message directing the user to ask their workspace admin to increase the limit.

— Describes behaviour but no config or command surface product docs

NEW SUPPORTED PLATFORM Rex terminal support

└──▷ BREAKING ON UPGRADE

! Extensions' `before_provider_headers` handlers no longer see the OpenRouter, NVIDIA NIM, and OpenCode attribution headers; returning `null` for one still removes it.

! The default GitHub Copilot model changed from `gpt-5.4` (retired by Copilot) to a new default — check your config if you pinned this provider.

! The default Xiaomi model changed to `mimo-v2.6-pro` for the `xiaomi` and Xiaomi Token Plan providers; `mimo-v2.5-pro` has left the catalog.

! generateSummary() and generateSummaryWithUsage() now ignore their output-cap parameter; summary requests use the model's output limit instead.

v1.4.0 NOTES ↗ LATEST STABLE

Adds `PI_CODEMODE_MAX_WORKERS` to queue parallel codemode workers, removes fixed summary output cap, and publishes a full `models.json` configuration reference.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.4.0 https://github.com/earendil-works/pi.git
# already have the repo? check out this version:
$ git checkout v1.4.0
```

Adds `PI_CODEMODE_MAX_WORKERS` environment variable to raise the codemode worker limit, so parallel scripts and in-process subagents queue instead of failing immediately.

Removes the fixed output cap on compaction and branch summaries — summary requests now use the model's full output limit; `compaction.reserveTokens` only

controls when compaction starts.

Adds a Models Configuration reference generated from the `models.json` schema, documenting every provider, model, model override, and compatibility field.

Adds provider-level `compat` fields `imageInput` and `sendImageOptInHeader` in `models.json` for proxies that serve another service's classifiers, now also applying to classifier models.

Adds `xhigh` thinking level for Grok 4.3 and Grok 4.5 on the `xai` provider.

└─▷ **BREAKING ON UPGRADE**

! generateSummary() and generateSummaryWithUsage() now ignore their `reserveTokens` parameter — callers that relied on it to bound summary output length will see that bound removed.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.307 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.307 adds a Code Review stop command, a new safety-event telemetry signal, an API-version migration command, and tightens self-hosted runner and settings behavior.

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

01 **Self-hosted runner checkout hook timeout** IMPROVED 85

Self-hosted runner checkout hooks now time out after 30 minutes by default, failing the session's setup; the limit is configurable via `--checkout-hook-timeout-min` (set to `0` for no limit).

Allow a long-running checkout hook more than the new 30-minute default when self-hosting a runner with complex environment setup.

```
$ claude --checkout-hook-timeout-min 60
```

— Default, flag, and override value all specified with example. v2.1.307

02 **@claude review stop/off/pause command** NEW 80

Adds `@claude review stop` (also `off` and `pause`) to Code Review, ending an `@claude review always` subscription on a pull request without starting a new review.

Stop an `@claude review always` subscription on a PR when continuous review is no longer needed, without triggering a new review.

```
$ @claude review stop
```

— Named command with runnable example and clear scope. v2.1.307

03 **claude_code.safety_event OpenTelemetry event** NEW 70

Adds the `claude_code.safety_event` OpenTelemetry event, emitted for each tool call that a permission rule, auto mode, a safety check, a hook, or the sandbox stopped, prompted, or sent back.

— Mechanism and triggers described but no setup steps given. v2.1.307

04 **`/claude-api api-version` command** NEW 70

Adds `/claude-api api-version` subcommand to move Claude API code to API version `2026-10-21`.

— Named command and version, but behavior detail is thin. v2.1.307

05 **`axScreenReader` setting scope restriction** BREAKING 65

Changes `axScreenReader` setting and `CLAUDE_AX_SCREEN_READER` in a settings `env` block to apply only from user, managed, or `--settings` settings.

— Names exact keys but no migration guidance for affected users. v2.1.307

06 **VSCode voice dictation timeout notice** NEW 60

[VSCode] Adds a notice on the microphone button when voice dictation stops by itself (15 seconds without recognized speech, or the 2-minute limit), stating why and that text so far is kept.

— UI behavior described with concrete timings, no exact navigation. v2.1.307

THINNER COVERAGE BELOW

07 **Retry for unparseable tool calls** IMPROVED 50

Improves recovery from unparseable tool calls: Claude Code now retries the call up to twice before ending the turn with an error.

— Describes behavior change but no user-facing control. v2.1.307

WAS THIS USEFUL?  

Zed

Feedback ⓘ v1.25.2 NOTES ↗ RANK

Zed is an open-source code editor that runs collaborative development workflows.

> Zed v1.25.2 adds new Bedrock AI models, lets you accept inlay hints directly in the editor, previews SVGs from the Project Panel, and trims binary size significantly across platforms.

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

01 **Inlay hint text edits applied inline** NEW 75

You can now apply inlay hint text edits (such as an inferred type) by double-clicking a hint, or by placing the cursor at the hint and running `editor: accept inlay hint`.

Accept an inlay hint suggestion (e.g. an inferred type) at the cursor position without reaching for the mouse.

📍 editor: accept inlay hint

— Exact command and interaction named, behaviour clear v1.25.2

02 **tab_switcher.show_file_preview setting** NEW 75

Adds `tab_switcher.show_file_preview` setting to control whether files are previewed in the editor while cycling through the tab switcher; defaults to `true`.

Disable file preview while tabbing through the tab switcher to avoid triggering file loads on every keystroke.

yaml

```yaml
tab_switcher:
  show_file_preview: false
```

— Named config key with default and override shown v1.25.2

THINNER COVERAGE BELOW

03 **SVG and CSS-in-JS syntax highlighting** NEW 55

Adds HTML syntax highlighting for `svg`, `mathml`, and `raw` tagged templates, and CSS syntax highlighting for template literals in css({ … }) objects, in JavaScript and TypeScript.

— Named template kinds and syntax but no steps to enable v1.25.2

04 **New Bedrock AI models added** NEW 50

Adds GPT-6 Astra, Claude Fable 5.1, and Claude Opus 5.5 to the Amazon Bedrock provider.

— Named models but no usage detail given v1.25.2

05 **Automatic unfolding on navigation** IMPROVED 50

Adds automatic unfolding of folded sections when jumping to a location via go to line, go to definition/declaration/implementation/type definition, symbol navigation, the outline and project symbol pickers, multibuffer location results, and navigation history.

— Lists navigation surfaces but no commands or keys v1.25.2

06 **SVG preview from Project Panel** NEW 45

Adds an Open SVG Preview option to the Project Panel's context menu for SVG files.

— UI path named, clear starting point v1.25.2

07 **Reduced binary size across platforms** IMPROVED 40

Reduces Zed's binary size by about 33% on macOS, 28% on Windows, and 7% on Linux, enabling use on more constrained systems.

— Concrete percentages but nothing for reader to act on v1.25.2

08 **ACP session notices and context compaction** NEW 30

Enables session notices and context compaction for external agents using ACP.

— Thin description, no mechanism or config shown v1.25.2

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.1317 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 biggest window changes were to its masking engine — per-kind Mask/Warn/Block actions, a never-mask allowlist, and new visibility panels — alongside adding Zoo Code as a supported agent target and fixing Claude Code's deferred tool loading and web.run conflicts.

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

01 Zoo Code support via autoImportSettingsPath config NEW

90

Adds Zoo Code (VS Code extension `ZooCodeOrganization.zoo-code`) as a supported agent target, injecting a magpie-managed provider profile on every extension start by routing through the `zoo-code.autoImportSettingsPath` setting, e.g. setting it to `<path-to-vscode-user-dir>/zoo-code-magpie-import.json` in `settings.json`.

Point Zoo Code at magpie's auto-generated import file so the extension picks up magpie as its active provider profile on every start.

json

```
{
  "zoo-code.autoImportSettingsPath": "<path-to-vscode-user-
dir>/zoo-code-magpie-import.json"
}
```

— Exact config key and a runnable settings.json snippet v0.1.1310

02 **Per-kind masking actions, allowlist, and visibility tools**

NEW

80

Each detected data category (e.g. `API_KEY`) can now be set to `Mask` (replaced with a placeholder), `Warn` (logged and sent as-is), or `Block` (held back with a reason returned to the agent). A `Never mask` allowlist lets specific literal values such as a team email or test key always pass through to vendors unmodified. A new `Try it` panel in the masking settings UI lets practitioners paste sample agent text and see exactly what a vendor would receive, without storing anything, and the masking card now also shows the exact lines the vendor received, sourced from the newest records.

— Explains mechanism and named actions but only a UI location, no command v0.1.1317

v0.1.1312

03 **Claude Code deferred tool loading via OpenAI Responses API**

NEW

65

Magpie's gateway now correctly routes Claude Code's deferred tool loading pattern, sending `additional_tools` items to OpenAI's Responses API (`LoadedInPlace`, issue `#2660`) instead of mishandling them.

— Names internal mechanism and issue but no user-facing step v0.1.1312

THINNER COVERAGE BELOW

04 **CLI sign-in re-checks only on file change** IMPROVED

55

CLI provider sign-in status is now re-queried only when its underlying files change, eliminating repeated ~0.5s CPU and ~200MB memory spikes that previously occurred each polling interval.

— Concrete numbers given but nothing for reader to act on v0.1.1316

05 **Bedrock Claude routed to its own Claude Code window** IMPROVED

55

A Claude model running on Bedrock is now routed to its own Claude Code window instead of sharing the 272K context window used by other instances.

— Specific context-window number but no config step given v0.1.1309

06 **Global AGENTS.md surfaced as Devin Instructions** NEW

55

The global `AGENTS.md` file is now surfaced as Instructions when using Devin, accessible via Library → Instructions.

— Names the file and UI path but no deeper mechanism v0.1.1309

07 **Codex web.run tool fixed alongside search history** IMPROVED

50

Magpie now supports Codex's `web.run` tool alongside a searched conversation history without incorrectly appending the hosted `web_search` tool, matching how Codex itself sends requests to the backend.

— Names both tools but gives no user action v0.1.1310

08 **Remote model list scoped to locally-picked decision models**

IMPROVED

45

A remote magpie instance's model list now carries only the decision models picked on that remote instance, preventing unpicked OpenRouter models from incorrectly appearing as local models.

— Describes the fix but no config or command surface v0.1.1316

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.102 NOTES ↗ RANK

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

> GitHub Copilot CLI v1.0.102 focuses on reliability and workflow polish: proxy-aware sandboxed networking, inherited MCP tool approvals for subagents, expanded plugin marketplace sources, and several smaller fixes to prompt mode, provider retries, long-running commands, and shell completion.

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

01 **Proxy-aware sandboxed networking** IMPROVED 65

Sandboxed network traffic now honors `NO_PROXY` and `ALL_PROXY` when inheriting host proxy settings, so commands run inside the sandbox respect existing proxy exclusions and routing.

Run a non-interactive prompt in a proxy-controlled environment so sandboxed commands respect your existing `NO_PROXY` and `ALL_PROXY` exclusions.

```
$ NO_PROXY=internal.corp ALL_PROXY=http://proxy.corp:3128 copilot
-p 'refactor the auth module to use OAuth2'
```

— Names both env vars and gives runnable example command v1.0.102

THINNER COVERAGE BELOW

02 **Auto model selection in prompt mode** IMPROVED 50

Prompt mode now honors managed Auto model selection for `-p` and piped stdin runs.

— Names exact flag and input mode affected v1.0.102

03 **MCP tool approval inheritance for subagents** IMPROVED 45

Subagents now honor custom and MCP tool approvals, avoiding duplicate permission prompts and allowing inherited workflows to run without re-approving tools already granted.

— Explains behavior but no concrete command or flag shown v1.0.102

04 **Plugin marketplace source improvements** IMPROVED 40

Plugin marketplaces now accept git-subdir sources and support long plugin descriptions.

— Names git-subdir sources but no usage example given v1.0.102

05 **Provider overload retry with backoff** IMPROVED 35

Overloaded-provider replies now retry with backoff and surface the provider's message and request ID.

— Describes mechanism but no config or command to invoke it v1.0.102

06 **Fish shell completion with option aliases** IMPROVED 35

Fish shell completion now works correctly when aliases include option aliases like `--effort`.

— Names the flag example but fix scope is narrow v1.0.102

07 **Long-running sandboxed command reliability** IMPROVED 30

Long-running sandboxed commands now continue reliably past 10 seconds, fixing earlier interruption behavior.

— Gives a before/after threshold but no further detail v1.0.102

WAS THIS USEFUL?  

Diagram Design

Feedback ⓘ changes since 2026-10-10 CODE ↗

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

Adds service blueprint visual type and architecture containment scopes to Diagram Design

└──▷ **GET THIS VERSION**

```
$ git clone --branch commits-2026-10-10
https://github.com/cathrynlavery/diagram-design.git
# already have the repo? check out this version:
$ git checkout commits-2026-10-10
```

.

└──▷ **TRY IT**

Validate a service blueprint diagram to catch silent layer-order or visibility-line violations before publishing.

```
$ python3 scripts/verify-service-blueprint.py <path/to/service-
blueprint.html>
```

Adds `service blueprint` visual type with a dedicated verifier (`scripts/verify-service-blueprint.py`) that enforces layer order, visibility-line placement, stage-grid alignment, and per-stage census checks using `data-layer`, `data-stage`, `data-name`, and `data-role="visibility"` declarations.

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

CLI: Workspace Trust, Workflow Plan Step Trees, and Mermaid Subgraphs (2026-10-20)

BLOG ↗ RANK

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

> Kiro CLI 2.33 adds workspace trust prompts and per-folder trust storage, step-tree visibility into workflow plans, and richer Mermaid rendering, while a separate release introduces persistent memory shared between the IDE and CLI.

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

01 Workspace trust prompt and per-folder trust storage NEW

80

Interactive V3 sessions in Kiro CLI now show a 'Trust this workspace?' screen before loading hooks, agents, MCP servers, steering, workflows, or language server configuration from a folder. Trust answers are saved per folder under `~/.kiro/workspace-roots/`; removing the `workspaceTrusted` key from the per-workspace file resets the choice so Kiro prompts again.

Reset workspace trust for a specific project so Kiro prompts you again on next launch.

> 📍 In the per-workspace file listed on the 'Trust this workspace?' screen (under ~/.kiro/workspace-roots/), remove the `workspaceTrusted` key and save. Kiro will prompt for trust again the next time you start a V3 session in that folder.

— Names exact config path and key used to reset trust.

CLI: Workspace Trust, Workflow Plan Ste…

THINNER COVERAGE BELOW

02 Workflow plan step trees IMPROVED

45

Workflow plan approvals now display a compact step tree, letting you expand individual steps to read details before execution begins.

— UI behavior described but no command or config to act on.

CLI: Workspace Trust, Workflow Plan Ste…

03 Mermaid subgraph rendering in chat IMPROVED

45

Mermaid flowcharts in chat now render subgraphs as titled frames, apply style-class colors, wrap node labels across up to four rows, and render `<br/>` as a second line.

— Visual rendering change only, nothing to configure or run.

CLI: Workspace Trust, Workflow Plan Ste…

04 **Model fallback command in CLI sessions** NEW 45

Running `/model fallback` in an active V3 session lets Kiro switch models automatically when the primary model is unavailable.

Set a model fallback in an active V3 session so the agent switches automatically when the primary model is unavailable.



— Only a usage example exists; mechanism not otherwise explained.

CLI: Workspace Trust, Workflow Plan Ste…

05 **Parallel grep scanning in V2 tool** IMPROVED 35

The V2 grep tool now scans files in parallel across CPU cores for faster searches on large projects.

— Performance claim with no benchmark, flag, or usage detail.

CLI: Workspace Trust, Workflow Plan Ste…

WAS THIS USEFUL? 👍 👎

OpenQodex

Feedback ⓘ openqodex@0.13.0 NOTES ↗ RANK

Open source code review that runs in your coding agent before you push. Scanners plus a separate AI reviewer, on your own login, no key.

> Across 13 releases OpenQodex built out its open-source, in-agent review pipeline end to end: a persistent interface-aware code graph served over MCP, framework-aware review briefs, per-project pinned scanner installation, an importable library API with `reviewChange`, and tighter GitHub Action, self-update and reviewer-backend controls.

└──▷ **WHAT SHIPPED · 13 FEATURES** most completely described first ? what's the number?

01 **Unified `review` command pipeline** IMPROVED 98

`openqodex review --agent` analyses a change, selects matching scanners, and prints a review brief for the coding agent; `review --finalize` evaluates agent findings without a model and writes the report as Markdown, JSON and SARIF, gated by `review.block_on_severity` in `.openqodex.yaml` (later overridable per-run with `scan --block-on-severity <severity>`). `openqodex review --all` scans every file in the repository, prioritising review starting from the most-called functions and files with the most scanner hits. `review <branch>` and `review '<PR link>'` review a branch or pull request outside current work, fetching and checking it out in a temporary folder and reviewing only what it added since its base; `--base` controls the base commit, falling back to the pull request's base (via `gh`), then `review.default_base`, then the remote's default branch; `review --finalize --run <id>` finalizes a single named run, whose ID the brief prints. Changes to a scanner's own settings or ignore file (e.g. `.gitleaksignore`, `ruff.toml`) are raised as a candidate the reviewer must clear; lines adjacent to a deletion count as changed, with each deletion point listed in the brief. As of 0.5.0, `review` performs a full single-command pipeline — snapshot, scan, code graph, isolated reviewer, answer verification, and final report — recording which agent completed the review in `reviewed_by` (report.json) and a run property in report.sarif; when no reviewer is available it names a fallback agent and saves unchecked candidates instead of showing raw scanner output as a review; push hooks trust only review records from `~/.openqodex/receipts/`, ignoring report files a branch carries under `.openqodex/`, with records over 30 days old removed by `init` / `update`; the git pre-push hook itself no longer scans or prints findings directly; `init` ends with a review of its own change, skippable with `--no-review`; the old `review --agent` / `review --finalize` agent rules are removed, leaving only `review` and `review --all`. The default terminal output became a short receipt (verdict, summary, one line per finding, paths to `report.html` / `report.md`) in 0.9.0; `--format markdown|json|sarif` restores the full report, `openqodex findings 1,3` prints named findings in full, and `report.html` renders a local, script-free diff page with findings anchored to lines, coverage, scanner list, blast radius

and redacted secrets. `--report-dir` (on `review` and `scan`) exits with code 2 before writing if the path resolves through a symbolic link. Config moved to `.openqodex/config.yaml`, adding `review.severity_threshold` (default `minor`) and `review.default_base`; the legacy root `.openqodex.yaml` is still read and `pr_review` is accepted as an alias of `review`.

— Names every flag and config key across the pipeline. openqodex@0.1.0 openqodex@0.2.0 openqodex@0.4.0 openqodex@0.5.0 openqodex@0.9.0

02 **Persistent, interface-aware code graph and MCP server** NEW

98

The review brief gains a code graph (0.2.0) — covering TypeScript, JavaScript, Python, Go and Ruby — showing which functions a change touches, their callers with exact call lines, files importing changed files, and removed functions still referenced elsewhere; cached under `.openqodex/graph/`, controlled by `graph.enabled: false` or `--no-graph`, plus `graph.budget_ms`, `graph.max_files`, `graph.max_file_bytes`. 0.10.0 adds cross-package caller resolution across pnpm, npm and yarn workspaces (via each file's nearest `tsconfig.json` and Python `src` roots), annotates every caller with a confidence level — `certain`, `likely`, or `possible` — naming the convention behind it, documents every truncation, persists the graph in `.openqodex/graph/` (`graph.max_cache_mb` default 512MB, `graph.max_heap_mb`), exposes each cached build via git ref `refs/openqodex/graph/<tree>`, writes full graph data into `.openqodex-review/graph/`, runs the graph on manifest-only changes, and ships hidden preview commands `openqodex graph build|status|search|symbol|callers|callees|importers|changes|unknowns|explain|capabilities` (`--json`). 0.11.0 promotes these to the public `openqodex graph` command, adding class hierarchy, method overrides, symbol usage, connectivity, reachability, file/folder definitions, dependencies and import cycles, plus `graph help`, `--budget-ms`, `--limit`, `--cursor`, `--tokens`; unanswerable questions exit with code 2. `openqodex mcp` serves these questions to coding agents as MCP tools over stdio, with `graph_refresh`, cursor-based pagination for `graph_changes` / `graph_impact`, and a concurrency limit of four queued builds; `init` registers this MCP server in `~/.claude.json`, `~/.cursor/mcp.json`, `~/.codex/config.toml` and the Cline CLI settings file (or, with `--project`, writes `.mcp.json` / `.cursor/mcp.json` / `.codex/config.toml` in the repo), with `init --no-mcp` / `init --mcp` to omit or restore it. Possible callers are shown at most 20 inline and interface-based calls keep at most 32 possible implementations. 0.12.0 advances the plugin interface to version 4 (Express plugin to version 7); the first build after upgrading reparses every file, discarding kept facts in `.openqodex/graph/`.

— Detailed mechanism, flags, limits and config keys throughout. openqodex@0.2.0 openqodex@0.10.0 openqodex@0.11.0 openqodex@0.12.0

03 **Per-project scanner selection and pinned installation**

IMPROVED 90

`openqodex scan` runs scanners only (for hooks, pre-commit and CI); `openqodex doctor` shows readiness with `--install`, and `--offline` skips `osv-scanner` / `semgrep` and disables downloads; custom scanners can be added by GitHub link in `.openqodex.yaml`, activated only after `openqodex trust` approves them. Thirteen built-in scanners activate only when relevant and report only changed-line findings, downloading on first use at pinned versions. 0.9.0 makes selection per-project: brakeman only in Rails apps (via Gemfile / `Gemfile.lock`, `config/application.rb` or `bin/rails`), rubocop Rails cops only in Rails apps, oxlint React/accessibility/Next.js rules only in matching projects, ruff Django/FastAPI/Airflow rules only in those projects; `doctor --install` (repository-scoped) installs only what's needed minus `scanners.disable` / `review.paths.exclude`, printing a reason each, while `doctor --install --all-scanners` restores installing everything. Dependencies are pinned and upgraded: `osv-scanner` to `2.6.0` (adds `bun.lock`, `uv.lock`, `pdm.lock`, `pylock.toml`, NuGet lockfiles, collapses aliased advisories), `oxlint` to `1.86.0` (installed from its GitHub release, sha256-verified, no npm dependency), and `semgrep` / `bandit` / `brakeman` / `rubocop` installed from lock files pinning every dependency with its sha256. 0.12.0 adds preinstallScanners({ installRoot, require }) to verify pinned scanners and runtime, `doctor --install --all-scanners --require-all` (exits 2 when any missing), createToolResolver(installRoot), and `scratchRoot` on `runScanners` isolating each run; it also reorders scanner candidates by severity, then scanner, file, line, rule and message, breaking stored/compared candidate IDs. 0.13.0 bounds glob evaluation for `review.paths.exclude`, `review.disabled_rules`, a lens, or a custom scanner's `paths`, refusing globs over 1,024 characters, and records the sha256 hash and byte size of each semgrep registry rule pack in `scan.json`.

— Exact versions, hashes and per-framework scanner rules named. openqodex@0.1.0 openqodex@0.9.0 openqodex@0.12.0 openqodex@0.13.0

---

04 **Importable library API with `reviewChange`** NEW 90

OpenQodex can be imported as a library, exposing the scanners, code graph, lenses, config parser and renderers with types. reviewChange(input, reviewer, options) reviews a change against the host's model in a server profile, producing checked findings, coverage, a model completion record (contract `openqodex-model-review-1`), and a report in Markdown, SARIF and JSON; options include `previousReviewedSha` (review only changes since the last review), `scopes` (isolate change/snapshot/scanners/graph/tools/context into named folders), and `context` (supply lessons/comments/summaries/notes/earlier findings as quoted data). An `authorize` budget callback fires before every model call with `requestChars`, and `deadlineMs` sets a hard stop. The library also exports `isFixturePath` and `ruleClassFor`, lenses marked `security: true` carry that field when loaded, and `review.paths.exclude` now applies to both paths of a renamed file.

— Options, contracts and exported types all named explicitly. openqodex@0.12.0

05 Init: agent install targets and setup flow IMPROVED

90

`openqodex init` installs the skill and push gate into Claude Code, Codex CLI, Cursor and Cline (`--uninstall` to remove), alongside `openqodex hook install` and a GitHub Action plus pre-commit hook for CI. It creates `.openqodex/config.yaml` and `.openqodex/custom-instructions.md` for team-wide commit, injecting a subagent review instruction into each agent's instruction file (custom instructions carried verbatim into every brief). `--no-repo` skips (and `--uninstall` removes) the team review section in `CLAUDE.md` / `AGENTS.md` ; a per-user launcher installs in `~/.openqodex/bin/` on every user-scope `init` , with the active version controlled by `~/.openqodex/runtime/current` ; user-scope `init` also adds Claude Code rules allowing `review --agent` , `review --finalize` , their `--all` / `--offline` forms, and `guide` to run unattended, removed by `init --uninstall` , which also removes update state and `~/.openqodex/config.yaml` when unchanged. 0.9.0 adds `CLAUDE_CONFIG_DIR` and `CODEX_HOME` environment-variable awareness, `--hook none` and `--no-repo` flags, `init --project` to write agent files inside the repository, and `init --dry-run` to print one line per scanner that would download (e.g. `brakeman: Rails app in backend/`) without downloading anything; `init --yes --agent <host>` is the canonical one-line install (`npx -y openqodex@<version> init --yes --agent <host>`). `init` consolidates prompts into a single 'Write these files?' confirmation after displaying a full plan, waits up to two minutes for a needed scanner to finish downloading before the first review, and writes a global instruction-file line (e.g. to `~/.claude/CLAUDE.md`): 'Before any push, review the change with the openqodex skill.'

— Every flag and config path named across four releases. openqodex@0.1.0 openqodex@0.2.0 openqodex@0.3.0 openqodex@0.9.0

06 AI reviewer backend selection and runtime controls IMPROVED

85

`--reviewer auto|claude|codex|cursor` and `reviewer:` in `~/.openqodex/config.yaml` select the code-review agent; `--reviewer codex` runs the full review with `codex exec` inside a read-only, network-isolated sandbox, and `--reviewer auto` picks Codex automatically when running inside Codex or when it's the only reviewer installed. `reviewer_web: on` grants the reviewer Claude Code's web tools (off by default in 0.5.0, then on by default from 0.6.0 with `reviewer_web: off` to disable). `--timeout <seconds>` (default 600) caps reviewer runtime; 0.13.0 raises this to 30 minutes for a change/branch/PR and 2 hours for `--all` , with `--timeout` still setting a custom cap and values above 2147483 seconds refused. `review.instructions_files` lets repository owners choose convention files quoted in review briefs, with their presence or absence recorded. The library's `reviewChange` adds `secondReviewer` to run a second model on the same brief and merge both

results, and `instructions` to pass repository owners' rules to the reviewer, with each finding carrying the reviewer's `suggestedChange`.

— Flags and config keys given; underlying model routing logic sparse. `openqodex@0.5.0` `openqodex@0.6.0` `openqodex@0.13.0` `openqodex@0.12.0`

---

07 **Review report provenance, timing and telemetry** NEW

85

`rounds.json` records each correction round's failed checks, the reviewer's answer and the changed line ranges sent (also returned by `reviewChange` as `result.rounds`); `usage.json`, written by `review`, records each reviewer round's model, tokens and cost. `result.scan`, added to `reviewChange`'s result, exposes the full scan record — candidates, each scanner's summary and rule packs, `selection` and `trail` — without secrets' fingerprints. `report.json` and `scan.json` gain a `timings` field recording `change_ms`, `scan_ms`, `graph_ms`, `brief_ms`, `reviewer_ms`, `finalize_ms` and `total_ms` (also `result.timings` via the exported `RunTimings` type); `scan.json` also gains `selection` and `trail` fields recording which scanners were chosen and why and the fate of every finding, plus per-scanner `stages`, `resolveMs` and `files` fields. Run files now redact secrets found as JSON keys in `submission.json`, not only as values.

— Every file and field named; mostly data shape, not a workflow. `openqodex@0.13.0`

---

08 **GitHub Action AI review and config controls** BREAKING

80

`ANTHROPIC_API_KEY` triggers a full Claude Code AI review inside the Action alongside existing scanners, posting findings to code scanning and a report to the job summary; input `review` (`auto` / `off` / `required`) controls when it runs, `claude-code-version` pins the exact Claude Code version installed, and outputs `reviewed`, `review-status`, `reviewer` report what ran. `version` and `claude-code-version` accept exact SemVer only; `latest`, a range, a path or a `file:` package now fail the step. When an AI review doesn't complete, the Action falls back to running scanners, and custom review instructions are read from the base branch so a pull request can't supply its own. 0.5.0 adds `config-from: head` (use the pull request's own config instead of the now-default base-branch config), `fail-on-tool-error: true` (fail the job on scanner install/tool failure instead of only warning), and `block-on-severity` (a gate the pull request's config cannot weaken), plus a `status` output set to `tool-failed` on scanner failure; a wrong `config-from` or `block-on-severity` value now fails the step, and a tool failure (exit 2) no longer fails the job by default unless `fail-on-tool-error: true` is set.

— Every input/output named; default-behaviour change needs care. `openqodex@0.7.0` `openqodex@0.5.0`

---

09 **Self-update engine and rollback** IMPROVED

77

`openqodex update` adds `--now`, `--rollback`, `--off`, `--on` and `--status` subflags; a background job installs releases at least 24 hours old whose npm provenance was signed by the release workflow, at most once per day, after a review/scan/push check through the launcher. Auto-update is disabled via `update: off` in `~/.openqodex/config.yaml`, `OPENQODEX_AUTO_UPDATE=0`, `--offline`, or CI environments; `doctor` shows the update state, and `doctor` / `review` / `scan` warn when a pinned npx version is behind the newest seen. 0.9.0's `update --status` lists every release notice between two versions (data leaving the machine, push-blocking behaviour, reviewer configuration), with back-filled notices for `0.5.0` and `0.6.0`; `update --rollback` now writes `skip_version: <version>` to `~/.openqodex/config.yaml` instead of disabling updates entirely (installations on 0.8.1 or earlier still disable updates entirely, since they can't read `skip_version`); the daily auto-update withholds releases that change the agent-contract or config-format version, requiring an explicit `openqodex update`.

— Config keys and env vars named; rollout mechanism detailed. openqodex@0.3.0 openqodex@0.9.0

---

10 **Framework route and component awareness in review** NEW

70

The review brief lists framework routes reaching changed code: Django and Rails routes (full paths, route names, routes left without a handler, rendered templates/views, changed model migrations, and tests that call, request or name the changed code), plus plugins for Express, React, Next.js, FastAPI and Go `net/http` surfacing each route's full path and handler, routers and middleware, components and their rendered children, hooks, and tests that request a route or render a component. Review packets gain `implementers/` and `references/` pages for each touched symbol, and `unknowns.json` listing framework gaps alongside the graph's own unknowns; the framework plugin API reaches version 3, letting a plugin share a single file walk across Express, React and Next.js readers. 0.13.0 adds Flask detection: `scan.json` names Flask among a project's frameworks when `requirements*.txt`, `pyproject.toml`, or Pipfile lists `flask`.

— Lists concrete framework surfaces but light on detection mechanism. openqodex@0.11.0 openqodex@0.13.0

THINNER COVERAGE BELOW

11 **Guide and issue-reporting commands** NEW

56

`openqodex report '<what went wrong>'` interactively files a GitHub issue, showing the exact issue text and requiring explicit confirmation before sending; code, paths, filenames and secrets are never included. `guide skill` prints the review procedure of the running version, decoupling the skill and instruction files from procedural content, with commands written for the launcher when it started the process. `guide privacy` prints the privacy policy, listing every network call made,

backed by `docs/privacy.md` documenting what OpenQodex collects (nothing). `guide for-agents` prints the AI-agent documentation offline.

— Commands named but each described in a single line. `openqodex@0.2.0` `openqodex@0.3.0` `openqodex@0.6.1` `openqodex@0.13.0` `openqodex@0.9.0`

12 **Demo repository generator** NEW 35

`openqodex demo` builds a small repository with planted bugs and scans it.

— Single-line description; no detail on planted bugs. `openqodex@0.1.0`

13 **Codex and Cursor plugin packaging** NEW 30

A Codex plugin is added in `plugins/codex/` for the OpenAI plugin directory, and `.cursor-plugin/plugin.json` makes the repository a Cursor plugin.

— Two files named; no further mechanism given. `openqodex@0.6.1`

└──▷ BREAKING ON UPGRADE

! The plugin interface is now version 4 and the Express plugin version 7; the first build after upgrading parses every file again, discarding kept facts in `.openqodex/graph/`.

! Scanner candidates are now ordered by severity, then scanner, file, line, rule and message; the numbering of candidates with the same severity changes once with this release, breaking any stored or compared candidate IDs.

! `openqodex update --rollback` no longer disables updates; it now writes `skip_version: <version>` to `~/.openqodex/config.yaml`. Installations running 0.8.1 or earlier that roll back will still have updates disabled (those versions cannot read `skip_version`).

! `doctor --install` now installs only scanners the repository's files require rather than all scanners; use `doctor --install --all-scanners` to restore the previous behaviour.

! The `review` command's default terminal output is now a short receipt rather than the full report; use `--format markdown`, `json`, or `sarif` to get the full report as before.

! `--report-dir` paths that resolve through a symbolic link now cause `review` and `scan` to exit with code 2 before writing anything.

! The daily auto-update now withholds releases that change the agent-contract or config-format version; upgrading to such a release requires explicitly running `openqodex update`.

! The Action's `version` and `claude-code-version` inputs now accept exact SemVer release versions only (e.g. `0.6.1`); supplying `latest`, a version range, a path, or a `file:` package now fails the step.

! The GitHub Action now reads pull request config from the base branch by default; the pull request's own config file is ignored unless `config-from: head` is set.

! A wrong `config-from` or `block-on-severity` value in the GitHub Action now fails the step.

! The git pre-push hook no longer scans or prints scanner findings.

! The push hooks trust only `~/.openqodex/receipts/` for review records; report files carried under `.openqodex/` in a branch are no longer trusted.

! The older `review --agent` and `review --finalize` agent rules are removed from the skill, agent rules, and team section; only `review` and `review --all` remain.

! A tool failure (exit 2) in the GitHub Action no longer fails the job by default; it now sets `status: tool-failed` and shows a warning unless `fail-on-tool-error: true` is set.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Databricks Omnigent

Feedback ⓘ announced · Oct 22, 2026

BLOG ↗ RANK

Open-source meta-harness that orchestrates Claude Code, Codex, Cursor, Pi and custom agents behind one interface, with policy controls, OS-level sandboxing and live session sharing.

> Omnigent's only release this window adds a Pull Requests panel directly inside coding-agent sessions, letting developers review PR context, CI results, and diffs without leaving the conversation.

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

01 **Pull Requests panel in coding sessions** NEW 81

A new Pull Requests panel in the workspace rail shows the PR description, CI check results, inline comments, and file diffs alongside the active agent session, for GitHub, GitLab, and Azure DevOps. It authenticates via `AZURE_DEVOPS_EXT_PAT` or `az login` for Azure DevOps, `gh` (GitHub CLI) for GitHub, and `glab` for GitLab as execution-host credentials. A PR indicator in the composer's workspace bar opens the panel when a pull request is associated with the session, a request selector lets users switch between multiple pull/merge requests tied to one session, and the panel refreshes automatically when an agent turn ends and polls for updates while open and the app is in the foreground.

— Names exact credentials, providers and refresh behavior; no UI screenshot or limits given

launch-20261022-4f50bdc3

WAS THIS USEFUL? 👍 👎

Arcade

Feedback ⓘ snapshot-20261022 NOTES ↗ RANK

Arcade provides an actions runtime that lets AI agents securely call external services with authorization and governance controls.

> Arcade shipped a cross-client agent plugin for connecting coding agents and desktops to its hosted gateway, alongside Helm chart updates for cloud-native credentials, schema validation, API pagination, and several tracing-related secret-leak fixes.

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

01 Agent plugin across coding agents and clients NEW

90

Arcade's plugin connects coding agents and desktop clients to apps via its hosted gateway. Install in Claude Code via `claude plugin install arcade --marketplace ArcadeAI/arcade-plugin` (requires Claude Code 2.1.292+), then `/reload-plugins`, `/mcp` to sign in, and `/arcade:try-arcade` to verify. In Codex via `codex plugin marketplace add ArcadeAI/arcade-plugin` and `codex plugin add arcade@arcade`, verified with `$arcade:try-arcade`. In Cursor via Team Marketplace (Dashboard → Plugins & MCPs → Team Marketplaces) or a local clone into `~/.cursor/plugins/local/arcade-plugin`, verified with `/try-arcade`. In VS Code via `Chat: Install Plugin From Source` pointed at `https://github.com/ArcadeAI/arcade-plugin`, requiring `chat.plugins.enabled`, verified with `/arcade:try-arcade`. In GitHub Copilot CLI via `copilot plugin marketplace add ArcadeAI/arcade-plugin` and `copilot plugin install arcade@arcade`, verified with `/try-arcade`. In Claude Desktop (paid plan required) via Customize → Plugins → Add → Add marketplace → Add from a repository using `ArcadeAI/arcade-plugin`. Generic MCP clients can connect to a remote HTTP MCP server at `https://api.arcade.dev/mcp/arcade`. After install, apps are connected by asking the agent (e.g. `Connect Google` or `Show my connected apps`), which triggers browser-based OAuth separate from gateway sign-in.

Install and verify the Arcade plugin in Claude Code to connect your coding agent to Arcade's hosted gateway without needing an API key.

```
$ claude plugin install arcade --marketplace ArcadeAI/arcade-plugin
```

Install the Arcade plugin locally in Cursor for an individual plan or a Teams/Enterprise plan with `Allow Local Plugin Imports` enabled.

```
$ git clone https://github.com/ArcadeAI/arcade-plugin
~/.cursor/plugins/local/arcade-plugin
```

— Exact install commands and verification steps for every supported client. product docs

02 **Helm chart values for cloud credentials and Postgres** NEW

90

New Helm values `deployments.network.egress.allow`, `engine.serviceAccount.annotations`, `engine.podLabels`, `engine.podAnnotations`, `engine.extraVolumes`, and `engine.extraVolumeMounts` support EKS IRSA, GKE Workload Identity, and file-based cloud credentials in place of static keys; all default to empty so existing installs are unaffected. A new `storage.postgres.url` Helm value (set via `engine.extraEnv`) configures the Postgres connection string.

Enable GKE Workload Identity for the engine so it assumes a cloud identity via a service-account annotation instead of static credentials.

```yaml
engine:
  serviceAccount:
    annotations:
      iam.gke.io/workload-identity-pool: 'my-project.svc.id.goog'
  extraVolumes: []
  extraVolumeMounts: []
```

Supply a Postgres connection string via the Helm chart instead of baking it into a static secret.

```yaml
engine:
  extraEnv:
    - name: STORAGE_POSTGRES_URL
      value: 'postgres://user:password@db-host:5432/arcade?
sslmode=verify-full'
```

— Named config keys with runnable YAML snippets for each use. snapshot-20261022

03 **Pagination for GET /v1/formatted_tools** IMPROVED

85

`GET /v1/formatted_tools`, which supports `openai`, `anthropic`, and `mcp` formats, now paginates before formatting, keeping response time flat for large tool catalogs.

— Exact endpoint, method, and supported formats given. snapshot-20261022

04 **Secrets no longer leaked into traces** IMPROVED 65

Engine trace spans no longer export Authorization, Cookie, API-key, or internal tool-secret header values to the tracing backend. Redis command arguments are also no longer recorded in traces, preventing secrets passed as Redis values (API keys, tokens) from leaking to the tracing backend.

— Named headers and data types, but no reader action required. snapshot-20261022

05 **Support image changes in Helm chart** BREAKING 60

The default `supportImages.curl` image changes to a `cgr.dev` image; allowlist that registry or override the `supportImages.curl` value before upgrading. `supportImages.busybox` is deprecated and will be removed in chart 1.13, so migration away from it should be planned.

— Named config keys and chart version give a clear migration path. snapshot-20261022

THINNER COVERAGE BELOW

06 **Static-worker MCP-Apps resources visible by default** BREAKING 45

Static-worker MCP-Apps resources are now visible on every gateway connection by default, removing the internal allowlist flag that previously hid them, with no toggle to revert.

— Behavior change described but the removed flag itself isn't named. snapshot-20261022

07 **Explicit error on tool store read failure** IMPROVED 35

Tool-listing endpoints and MCP discovery now return an explicit error (not an empty tool list) when the tool store cannot be read.

— Describes the change but names no endpoint or error format. snapshot-20261022

└──▷ **BREAKING ON UPGRADE**

! The default `supportImages.curl` image changes to a `cgr.dev` image; allowlist that registry or override the `supportImages.curl` value before upgrading.

! `supportImages.busybox` is deprecated and will be removed in chart 1.13; plan migration away from it.

! Helm values with wrong types are now rejected by schema validation at install/upgrade/template time — fix any misconfigured values before upgrading. See Upgrade notes.

! Static-worker MCP-Apps resources are now exposed on every gateway by default with no toggle to revert; the internal allowlist flag is gone.

WAS THIS USEFUL?  

Feynman

Feedback ⓘ v0.5.52 NOTES ↗ RANK

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

> Feynman added a new web search provider option and improved reliability of background research subagents across restarts.

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

01 Linkup search provider support

NEW INTEGRATION NEW

75

Feynman adds Linkup as a new search provider, configured via `feynman search set linkup <key>`. Each search performed through Linkup consumes Linkup credits.

Register a Linkup API key so Feynman routes web searches through Linkup rather than the default providers.

```
$ feynman search set linkup <key>
```

— Names exact command and billing behavior product docs

THINNER COVERAGE BELOW

02 Background subagent completion persists across restart

IMPROVED

40

Background research subagents now retain their completion notice across a restart, surfacing the result in the next session that opens them.

— Explains behavior but no config or command given product docs

v0.5.52 NOTES ↗ LATEST STABLE

Parallel subagents can now set their own `cwd` and `model`; most search providers support a `<provider>BaseUrl` gateway override.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.5.52 https://github.com/Companion-Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.52
```

> Allows each child of a `parallel` call to set its own `cwd` and `model`, enabling mixed-model parallel research runs.
>
> Supports routing most search providers through a custom gateway via a `<provider>BaseUrl` config setting, with API keys dropped on cross-site redirects.
>
> Subagents no longer inherit `APPEND_SYSTEM.md`, isolating system-prompt customisations to the main session.

WAS THIS USEFUL? 👍 👎

LangChain Deep Agents

Feedback ⓘ deepagents==0.7.30

NOTES ↗ RANK

Deep Agents provides an agent harness with planning, filesystem tools, subagents, and context management.

> Deep Agents 0.7.30 extends AsyncSubAgent with webhook notifications and non-blocking follow-up queuing for concurrent delegation.

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

01 AsyncSubAgent webhook notifications and non-blocking queuing

NEW 72

`AsyncSubAgent` gains `notify_parent` and `webhook` parameters, allowing an external service to be notified via HTTP callback when a long-running async sub-agent finishes instead of polling. Async sub-agent follow-ups can now queue without interrupting active work, enabling non-blocking concurrent delegation.

Trigger an HTTP callback when a long-running async sub-agent finishes, so an external service (e.g. a CI pipeline or Slack bot) is notified without polling.

python

```python
from deepagents import AsyncSubAgent

sub = AsyncSubAgent(
    model="openai:gpt-6-astra",
    system_prompt="Analyse the repository and produce a security summary.",
    webhook="https://hooks.example.com/deepagents/results",
    notify_parent=True,
)
```

— Names exact parameters and shows runnable code, but no mechanism detail on queuing.

deepagents==0.7.30

WAS THIS USEFUL? 👍 👎

Graphicle AI Graphicle

Feedback ⓘ v0.1.4 NOTES ↗

RANK

Code graph and memory for your coding agents, in one Rust program. For Claude Code, Codex, Cursor and any MCP client.

> Graphicle's three releases this window overhaul the `graphicle context` retrieval pipeline with notes-first ranking, identifier routing, query-focus extraction and identifier-arm scoring, add Windows preview binaries and installer support, trim Claude Code's default MCP tool set from 25 to 5, and add OS certificate store and custom CA trust for TLS connections.

└──▷ **WHAT SHIPPED · 20 FEATURES** most completely described first ? what's the number?

01 **Windows x86-64 preview binaries and installer** NEW INTEGRATION

NEW

93

Adds Windows support: `graphicle-<version>-x86_64-pc-windows-msvc.zip` release archives (preview) containing `graphicle.exe` and `graphicle-graph.exe` statically linked against the C runtime; an `install.ps1` PowerShell installer for Windows PowerShell 5.1 or PowerShell 7 that installs both programs into `%LOCALAPPDATA%\Programs\Graphicle\bin`, verifies SHA-256, adds the folder to the user `PATH`, and requires no administrator rights; and a Windows-specific data directory at `%LOCALAPPDATA%\Graphicle` with Claude Desktop (including Microsoft Store build) and VS Code settings located via their Windows paths.

— Full installer mechanism, paths and verification named. v0.1.1

02 **Notes-first ranking and staleness dates in context** NEW

88

Adds a `notes_first` option (off by default), configurable via `notes_first: true` in `context`, `graphicle context --notes-first`, or `graphicle mcp --set notes_first=true`, which enables note-prioritised ranking, one-excerpt-per-file deduplication, and per-excerpt staleness dates. A separate `source_age` option controls the per-excerpt staleness dates ((changed <date> UTC) and `may be stale: <file>`) independently of `notes_first`.

— Names config keys, flags and exact staleness formats. v0.1.3

03 **Claude Code tool set trimmed via setup --tools** BREAKING

88

`graphicle setup` now gives Claude Code five tools (`query_graph`, `get_node`, `get_neighbors`, `recall_project`, `read_project_source`) by default via

`graphicle mcp --tools essentials` ; `setup --tools all` restores the previous 25-tool behaviour. This is a breaking change from the prior default of 25 tools.

— Names exact tool list, flag and prior default. v0.1.1

04 **Identifier route for direct code lookups** NEW 86

`graphicle context` and `recall_project` add an identifier route: questions that are pure identifier lookups now return `path:line` definitions and references from the code graph instead of full retrieval excerpts, controlled by `identifier_route: false` , `graphicle context --no-identifier-route` , or `graphicle mcp --set identifier_route=false` .

— Exact flags and resulting behaviour described. v0.1.3

05 **Identifier-arm file scoring in context** NEW 86

Adds an `identifier_arm` option (`'auto'` by default) that scores files by the identifiers and paths a code question names, fused as the `identifiers` arm and weighted via the `arm_weights` key `identifiers` ; configurable via `graphicle context --identifier-arm` or `graphicle mcp --set identifier_arm=false` . The packet's `identifier_arm` field lists the names and paths used.

— Names config key, flag and scoring mechanism. v0.1.3

06 **Next-ranked-files listing via more_files** NEW 86

Adds a `more_files` option (default 10 for code questions, 0 otherwise; range 0–40) that appends paths of the next ranked files not quoted in excerpts, configurable via `graphicle context --more-files N` ; printed under Next ranked files and echoed in the packet's `more_files` field.

— Exact default, range, flag and field named. v0.1.3

07 **Prompt-context hook for Claude Code** NEW 86

Adds a `graphicle setup --prompt-context` flag (off by default) that installs a Claude Code `UserPromptSubmit` hook running `graphicle context -` (reading the question on standard input) and injecting up to 8,400 characters of context before each code question; `graphicle disconnect` removes it.

— Names flag, hook type and removal command. v0.1.3

08 **Per-call test-file ordering via include_tests** NEW

86

Adds an `include_tests` argument to `context` (via `graphicle call context --arguments` and the HTTP API) to control test-file excerpt ordering per call: `"auto"` (default), `false` (always after), or `true` (the 0.1.0 order); the packet's `test_files` field reports what was applied.

— Names argument, values and reporting field. v0.1.1

09 **Certificate trust and TLS error diagnostics**

NEW SUPPORTED PLATFORM IMPROVED

85

Graphicle now trusts the OS certificate store (Windows certificate store, macOS keychains, Linux system bundle) in addition to bundled roots, and supports `SSL_CERT_FILE` and `SSL_CERT_DIR` environment variables to supply custom CA certificates, overriding the system trust store when set. TLS connection error messages now report the specific cause (e.g. `invalid peer certificate: UnknownIssuer`) and actionable remediation hints — install the CA in the system trust store or point `SSL_CERT_FILE` at a PEM file — across `graphicle ask`, the graph engine model, pull request triage, and URL fetch requests.

— Names exact env vars, cert stores and error message format. v0.1.4

10 **Query-focus extraction in context retrieval** NEW

84

Adds a `query_focus` option (`'auto'` by default) that strips instruction words from a code question and focuses retrieval on marked content (e.g. `<issue>` tags or `--- BEGIN ISSUE ---` / `--- END ISSUE ---` delimiters), configurable via `graphicle context --query-focus`; the packet's `query_focus` field reports what was marked or left out.

— Names flag, markers and packet field. v0.1.3

11 **Guard hooks opt-in via setup --guards** BREAKING

82

Adds a `setup --guards` (and `connect --guards`) flag to opt in to guard hooks that point Claude Code's searches to graph tools; running `setup` without `--guards` now removes any previously installed guard hooks, a breaking change from prior behaviour.

— Names flags and describes behaviour change. v0.1.1

12 **Automatic new-release notifications** NEW INTEGRATION NEW

82

Adds automatic new-release notices that query GitHub's releases API once per day and print the new version, upgrade command, and release URL to stderr when a newer release is available, plus a `GRAPHICLE_NO_UPDATE_NOTIFIER=1` environment variable and `graphicle config set update_notifier false` config key to disable them.

— Names env var, config key and cadence. v0.1.1

13 **Context packet size cap via --max-chars** IMPROVED 81

`graphicle context` adds a `--max-chars` flag, defaulting to 8,400 characters, to cap packet size — replacing the previous token-based default of 12,000 tokens × 4 characters.

— Exact flag, default and prior behaviour given. v0.1.3

14 **Incremental workspace import linking** IMPROVED 74

Workspace import linking is now incremental: after `graphicle graph update`, only projects that changed, whose export set changed, or whose linked-project set changed are relinked; the `graph_link` result reports relinked and reused projects.

— Describes mechanism and reporting but no numbers. v0.1.3

15 **Expanded language and template support in code graph** IMPROVED 63

Adds Jinja template support (`.j2`, `.jinja`, `.jinja2`) as document nodes in the code graph, and extends cross-repository member-call binding to PHP, TypeScript, JavaScript, Kotlin, and Objective-C receivers.

— Names extensions and languages but little mechanism. v0.1.1

THINNER COVERAGE BELOW

16 **Context mechanism switches exposed via MCP** IMPROVED 57

`code_first`, `query_focus`, `identifier_arm`, and `more_files` are exposed as MCP switches settable via `graphicle mcp --set` and echoed in `mechanisms.switches` of every packet.

— Names fields and command but no further mechanism. v0.1.3

17 **New hook installation subcommands** NEW 57

Adds `hook install` support for a `post-merge` hook alongside existing hooks, and a `pi guard install` subcommand to install a Pi guard extension.

— Two thin named subcommands grouped, no mechanism given. v0.1.1

18 **AST cache schema bump to version 9** BREAKING 57

The AST cache schema moves to version 9; the first build after upgrading to v0.1.1 will re-extract all files.

— States version number and migration consequence. v0.1.1

19 **Ollama requests switched to /api/chat** NEW INTEGRATION BREAKING

53

Ollama requests are now sent to `/api/chat` (with a fallback for older servers) instead of the previous endpoint.

— Names new endpoint but not the one it replaces. v0.1.1

20 **Graph benchmark JSON field labeling** IMPROVED 50

`graphicle graph benchmark` JSON output now uses Graphify's field names to correctly label its size-ratio figure.

— Minor labeling fix, thinly described. v0.1.1

└──▷ **BREAKING ON UPGRADE**

! The AST cache schema moves to version 9; the first build after upgrading to v0.1.1 will re-extract all files.

! `graphicle setup` now installs only 5 tools for Claude Code by default instead of 25; use `setup --tools all` to restore the previous behaviour.

! `graphicle setup` without `--guards` removes guard hooks that were installed by an earlier setup run.

! Ollama requests are now sent to `/api/chat` (with a fallback for older servers) instead of the previous endpoint.

v0.1.0 NOTES ↗ STABLE

Graphicle v0.1.0 debuts a local-first Rust code-graph engine with MCP server, workspace-wide recall, agent setup automation, and configurable project limits.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.1.0 https://github.com/Graphicle-AI/graphicle.git
# already have the repo? check out this version:
$ git checkout v0.1.0
```

Adds `graphicle graph <command>` to run the Rust code-graph engine, with opt-in additions `--precise`, `extract --docs`, `extract --routes`, `retrieve` and `serve --watch`.

Adds `graphicle sync` to maintain a local SQLite store of code, documents and agent sessions and build the code graph.

Adds `graphicle context` and `graphicle context --workspace` for cited recall within a token budget across one or many projects (`max_projects` up to 32).

> Adds `graphicle ask` for cited answers via a configured model (`graphicle config set model anthropic|openai|ollama`).
>
> Adds `graphicle setup` for one-command repository onboarding including Git hook installation and agent connection.
>
> +6 more

WAS THIS USEFUL? 👍 👎

openexecutive

Feedback ⓘ v0.6.4 NOTES ↗ RANK

AI-powered virtual executive team — a single coherent executive persona backed by 8 specialist agents (FastAPI + Next.js).

> OpenExecutive's v0.6.3 and v0.6.4 releases add workflow-driven email and calendar automation with injection-checked auto-replies, chat capabilities for workflows, opt-in Logfire request tracing, and a round of Learning-management, onboarding and attachment improvements.

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

01 Workflow-driven email and calendar automation NEW

70

Workflows can read the creator's mail and calendar as action inputs, send email as the creator either as a draft or sent immediately, and restrict outbound sending to recipients the creator pre-approved. An injection-checked delegation path lets a workflow answer email the same way Handle it for me would.

— Clear mechanism and scope but no exact API/flag names v0.6.4

02 Logfire request tracing NEW

65

Adds opt-in Logfire request tracing under the `observability` module for full request-level visibility.

— Names exact module and tool, opt-in behaviour, but limited mechanism detail v0.6.3

THINNER COVERAGE BELOW

03 Chat capabilities for workflows NEW

55

Action steps in workflows can now invoke the Executive's chat tools, and a delegation workflow can send a chat draft written in the user's voice on the user's tap.

— Describes mechanism but lacks concrete config or command names v0.6.4

04 Learning management improvements IMPROVED

55

Splits the Learning view by category in the UI, enables forgetting several Learning entries at once, and files each Learning miss under a finer cause category for more granular memory diagnostics.

— Groups three related Learning-module changes with some behavioural detail v0.6.3

05 New extension tool options in orchestrator IMPROVED

50

Adds `activity`, `private-row`, and `caller` options to extension tools in the orchestrator.

— Names three flags but doesn't explain what each does v0.6.3

06 **Drive file attachments in delegation workflow** NEW

50

Supports attaching up to 15 Drive files to a draft in the delegation workflow.

— Gives a concrete limit but little mechanism beyond that v0.6.3

07 **Onboarding and goal-setting improvements** IMPROVED

45

Department goals can now stand independently with an assigned owner, onboarding automatically selects Personal-or-Team with goals saved with owners, and initial setup configuration is drafted from the company's public website during onboarding.

— Groups several onboarding tweaks without deep mechanism detail v0.6.4

08 **Waiting-on replies view in briefing panel** NEW

40

Shows replies the user is waiting on in the Right now briefing panel.

— Names the panel but no further mechanism given v0.6.4

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ `langchain==1.4.7` NOTES ↗ RANK

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

> LangChain 1.4.7 adds support for streaming partial tool-call results and a configurable tracing endpoint.

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

01 **Tool-call-delta streaming support** NEW 50

Stream consumers can now read `tool-call-delta` events, enabling real-time streaming of partial tool-call results.

— Names the event type but no usage example `langchain==1.4.7`

02 **Custom tracing endpoint for LangChainTracer** NEW 50

`LangChainTracer` now supports an `address` parameter to specify a custom tracing endpoint.

— Names exact parameter and class but no example `langchain==1.4.7`

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ typescript/v1.22.0 NOTES ↗

RANK

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

> Strands Agents added a way to cancel a running agent mid-execution, useful for timeout handling or user-initiated stops.

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

01 **Mid-execution agent cancellation** NEW 65

A new agent.cancel() method stops a long-running agent mid-execution, useful in timeout handlers or when a user requests cancellation.

Stop a long-running agent mid-execution — useful in timeout handlers or when a user requests cancellation.

python

```
agent.cancel()
```

— Names exact method and use case but no mechanism detail python/v1.62.0

NEW INTEGRATION OpenAI Live provider (experimental, Python)

NEW INTEGRATION opencode model provider added to catalog

NEW INTEGRATION strands-you (You.com web tools plugin) added to integrations catalog

NEW INTEGRATION attenu-guard community intervention added to catalog

typescript/v1.22.0 NOTES ↗ LATEST STABLE

Adds swarm tool (Python + TypeScript), agent.cancel(), public Python middleware, invoke limits and usage accounting, baggage propagation for TypeScript tracing, and decision usage in invocation context.

└──▷ GET THIS VERSION

```
$ git clone --branch typescript/v1.22.0 https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout typescript/v1.22.0
```

> Adds agent.cancel() to stop the Python agent event loop mid-run.
>
> Adds swarm tool in TypeScript for multi-agent coordination.
>
> Adds swarm tool in Python for multi-agent coordination.
>
> Makes the Python middleware system public, enabling first-class middleware hooks in production agents.
>
> Adds enforce invoke limits and usage accounting across the whole agent tree.
>
> +2 more

WAS THIS USEFUL?  

vibe-trading

Feedback ⓘ v0.1.17 NOTES ↗ RANK

"Vibe-Trading: Your Personal Trading Agent"

> vibe-trading v0.1.17 adds daily paper trading against live markets, spending and decision caps for the live runner, strategy cloning/adaptation, new portfolio import and rebalancing tools, an mStock connector, support for custom OpenAI-compatible model endpoints, a shared multi-process research task board, and in-place editing of scheduled runs.

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

01 **Daily paper trading against completed bars** NEW 95

Adds the `paper_trade_step` agent and MCP tool, `vibe-trading paper create|list|show|history|advance`, `POST /api/paper-trade/step`, and `POST|GET /api/paper-accounts` (including `POST /api/paper-accounts/{id}/advance`) for daily paper trading against completed bars across `cn_a`, `hk`, and `us` markets. Ledgers are stored in `<runtime_root>/paper-trading/accounts.db`, and concurrent or stale advances are refused with HTTP 409.

— Full CLI, endpoints, storage path and conflict behavior named v0.1.17

02 **Live runner decision cadence and spending caps** NEW 90

Adds `GET` / `PUT /live/runner/settings`, `vibe-trading connector runner-settings`, and a Runtime → Model schedule UI to configure the live runner's decision cadence (default 15 minutes), a cap of 26 decisions and 1,000,000 reported tokens per broker per UTC day, and an optional USD spending limit.

— Endpoint, CLI, and exact numeric caps all given v0.1.17

03 **Strategy cloning and adaptation** NEW 90

Adds the `adapt_strategy` agent and MCP tool, `vibe-trading strategy-evidence adapt`, and `POST /api/strategy-adaptations` to clone a registered strategy to a new name, universe, date window, and position sizing. `strategy-evidence refresh --scope strategies` / `POST /api/strategy-adaptations/evidence` refreshes only the named strategies' evidence.

— Named endpoint, CLI, and refresh scope flag v0.1.17

04 **Custom OpenAI-compatible model providers** NEW 90

Adds `vibe-trading provider custom list|add|activate` and a custom-provider manager in Settings for registering and activating custom OpenAI-compatible

model endpoints. Keys go to the OS credential vault via the `[keyring]` extra, and `LANGCHAIN_MAX_TOKENS` caps output tokens on the OpenAI-compatible adapter.

Register a local OpenAI-compatible model endpoint (e.g. a self-hosted proxy) as a named provider and activate it for agent calls.

```
$ vibe-trading provider custom add
vibe-trading provider custom activate <profile-id>
```

— Runnable CLI example plus env var and extra named v0.1.17

05 **extraETF Investments CSV import** NEW 85

Adds `vibe-trading portfolio import-extraetf`, `POST /api/portfolio/imports/extraetf`, and a Portfolio → Manage accounts → Import extraETF Investments UI path to ingest an Investments CSV as a local holdings source, with EUR added as a display currency alongside USD, CNY, and HKD.

— CLI, endpoint and UI path all named v0.1.17

06 **Portfolio rebalance proposals** NEW 85

Adds `portfolio rebalance --snapshot-id`, `POST /api/portfolio/rebalance`, and portfolio_summary(snapshot_id=..., target_weights=...) to produce allocation proposals showing value changes for target percentages totalling 100%.

— CLI flag, endpoint and function signature given v0.1.17

07 **Snapshot-pinned portfolio analysis** NEW 85

Adds `snapshot_id` and `read_identity` return fields and a `snapshot_id` argument to `portfolio_summary`, plus `vibe-trading portfolio show --snapshot-id ID` and `GET /api/portfolio/analysis-context?snapshot_id=` to pin multi-step portfolio research to a single stored snapshot so results stay consistent even if prices refresh mid-session.

Pin a portfolio analysis to one snapshot so a multi-step agent workflow reads consistent holdings even if prices refresh mid-session.

```
$ vibe-trading portfolio show --snapshot-id <snapshot-id>
```

— CLI example and endpoint make it directly runnable v0.1.17

08 **Shared pixel-office research task board** NEW 75

Adds the `/pixel-office` Web page, `vibe-trading office` CLI, `GET /api/pixel-office`, `POST /api/pixel-office/actions`, and a `pixel_office`

agent tool sharing a transactional SQLite board at `pixel-office.sqlite3` under the runtime root for a shared manual research-task board across processes.

— Endpoints and storage path named, no worked example v0.1.17

09 **mStock read-only connector** NEW 65

Adds the `mstock-live-rest-readonly` connector profile for read-only GET-only access to mStock (India) holdings, net positions, and fund summaries, with `connector account` and `connector positions` gaining a `--connection-id` option.

— Profile and flag named but no full command shown v0.1.17

10 **In-place editing of scheduled runs** NEW 65

Adds `PATCH /scheduled-runs/{id}` and in-place editing on the Scheduled page to modify a job's prompt, cadence, timezone, end date, and delivery without deleting it; conflicting edits on a running job are refused with HTTP 409.

— Endpoint named, UI path only generally described v0.1.17

WAS THIS USEFUL?  

Mastra

Feedback ⓘ @mastra/core@1.79.0 NOTES ↗ RANK

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

> Mastra's headline changes this window are breaking updates to the agent controller session API and observability storage contracts (querySpans, thread-row queries), alongside new CLI trace search, Studio auth isolation, a Models fallback panel, Microsoft Teams channel support, and a one-roundtrip sandbox filesystem API.

└──▷ **WHAT SHIPPED · 14 FEATURES** most completely described first ? what's the number?

01 Multi-session agent controller support BREAKING

93

Mastra now supports many sessions per `resourceId` via an updated agent controller API: controller.getSession(sessionId), updated createSession() and deleteSession({ sessionId }) signatures, and new server routes under `/agent-controller/:controllerId/sessions/:sessionId/...`. This is a breaking change — controller.getSessionByResource(resourceId, scope) is removed, `scope` is removed from createSession(), deleteSession() now requires `{ sessionId }`, and the session() / session.create() signatures in `client-js` are updated accordingly.

Create multiple independent sessions for the same resource and address each by its own `sessionId`.

typescript

```
await controller.createSession({ id: 'run-a', resourceId: 'repo-1' });
await controller.createSession({ id: 'run-b', resourceId: 'repo-1' });

const sessionA = await controller.getSession('run-a');
await controller.deleteSession({ sessionId: 'run-b' });
```

— Exact methods, routes, and migration before/after given @mastra/core@1.79.0

02 CLI trace search with `mastra api trace query` NEW

86

The Mastra CLI gains `mastra api trace query`, a subcommand to browse and filter hosted traces from the terminal. It accepts a JSON query body with `timeRange`, `where`, `page.limit`, and field paths such as `durationMs`; results come back under `data.traces`, and the CLI reads project environment variables and targets the US observability endpoint by default, with other regions available via CLI target resolution.

Find completed traces whose root span took more than five seconds within a given week — useful for surfacing slow agent or workflow executions in CI or an ops shell.

```
$ npx mastra api trace query '{
  "timeRange": {
    "from": "2026-08-01T00:00:00.000Z",
    "to": "2026-08-08T00:00:00.000Z"
  },
  "where": {
    "op": "gt",
    "left": { "path": "durationMs" },
    "right": { "literal": 5000 }
  },
  "page": { "limit": 25 }
}'
```

— Runnable CLI example with exact query shape `product docs`

03 **Span and trace aggregation APIs for storage adapters** NEW

85

Adds an aggregateSpans() storage contract and the `span-aggregate` storage feature flag, enabling per-span analytics like p95 latency per tool or token/cost per agent over individual spans. Also adds aggregateTraces(), which groups by root span `entityId`, accepts `entityId` as a trace-scope `where` field, and is advertised via the new `trace-query-root-entity-id` storage feature for stores that support it.

Implement aggregateSpans() in a custom storage adapter to answer p95 latency or token-spend questions over individual spans.

typescript

```
class MyObservabilityStore extends ObservabilityStorage {
  getFeatures(): readonly ObservabilityStorageFeature[] {
    return [...(super.getFeatures() ?? []), 'span-aggregate'];
  }

  async aggregateSpans(plan: TrustedSpanAggregatePlan):
Promise<SpanAggregateResponse> {
    // e.g. { rows: [{ dimensions: { name: 'search' }, measures:
{ count: 42, 'duration.p95': 812 } }], truncated: false }
  }
}
```

— Concrete storage contract with implementation example `@mastra/core@1.79.0`

04 **Batch `experimentIds` filter in `getMetricBreakdown()`** NEW

84

Adds a batch `experimentIds` filter to getMetricBreakdown() so token usage and cost for many experiments can be fetched in one request, instead of one call per experiment.

Pull token usage across multiple experiments in a single observability query instead of one request per experiment.

typescript

```typescript
const result = await client.getMetricBreakdown({
  name: ['mastra_model_total_input_tokens',
'mastra_model_total_output_tokens'],
  aggregation: 'sum',
  groupBy: ['experimentId', 'name'],
  filters: { experimentIds: ['experiment-1', 'experiment-2'] },
});
```

— Exact method and code example provided `@mastra/core@1.79.0`

---

05 **Dedicated auth provider for Studio via `studio.auth`** NEW

75

Adds a `studio.auth` config key to assign Studio its own authentication provider, separate from `server.auth`, so Studio requests—including its API route calls—use independent credentials.

Isolate Studio behind its own auth provider so that Studio API calls don't share credentials with the main server.

javascript

```javascript
{
  server: {
    auth: mainAuthProvider
  },
  studio: {
    auth: studioAuthProvider
  }
}
```

— Named config key with working example `product docs`

---

06 **`textDisplay: 'hidden'` to suppress agent narration** NEW

75

Adds a `textDisplay: 'hidden'` config option that hides an agent's narration text from a thread while still allowing tool-posted messages, tool display, approval prompts,

error messages, and typing status to appear normally.

Suppress intermediate agent narration ('Running the tests now...') from a thread so only explicit tool-posted messages appear.

```yaml
textDisplay: 'hidden'
```

— Exact config flag with example usage `product docs`

---

07 **Sandbox filesystem API for one-roundtrip file operations** NEW

71

`WorkspaceSandbox` gains an optional `filesystem` property enabling one-roundtrip reads, writes, and listings instead of issuing a shell command per operation. The filesystem.walk() API also adds `includeGitignoreFiles` and `skipDirectoryNames` options for controlling directory traversal.

— Names exact properties and options, no code example `@mastra/core@1.79.0`

---

08 **`requestContext` added to channel resolver contexts** BREAKING

71

Adds `requestContext` to the `resolveResourceId` and `resolveThreadId` channel hook contexts, exposing per-message context set by handlers directly in resolvers. This is a breaking change: these contexts now require a `requestContext` field, so code that builds them by hand (e.g. tests calling a resolver directly) must pass requestContext: new RequestContext().

— Exact field and migration snippet named, no runnable example `@mastra/core@1.79.0`

---

09 **`bundler.pnpmWorkspaceSettings` for deployment workspaces**

NEW

70

Adds a `bundler.pnpmWorkspaceSettings` config key supporting `allowBuilds`, `onlyBuiltDependencies`, `ignoredBuiltDependencies`, `minimumReleaseAge`, `minimumReleaseAgeExclude`, `trustPolicy`, `trustPolicyExclude`, `trustPolicyIgnoreAfter`, and `blockExoticSubdeps` for generated deployment workspaces.

— Full list of sub-options named, no usage example `@mastra/core@1.79.0`

---

10 **Optional `threadId` param for thread ownership transfer**

IMPROVED

67

session.thread.transferOwnership({ toOwnerId, threadId }) gains an optional `threadId` parameter so a thread can be transferred to a new owner without first switching the session to that thread.

– Exact method signature named, no example `@mastra/core@1.79.0`

11 **Breaking changes to storage query contracts** BREAKING

65

The querySpans() storage contract drops the `model` and `provider` filters and removes `cost`, `inputTruncated`, `outputTruncated`, and `errorTruncated` from responses, and the thread-row query contract now requires `timeRange.traces` / `timeRange.threads` instead of the previous `traces.timeRange` field, so storage adapters must be updated accordingly.

– Names exact fields removed but no migration example Exact before/after field names, no broader explanation

`@mastra/core@1.79.0`

12 **`DURABLE_AGENT_OBSERVE_AGENT_MISMATCH` error for cross-process observe**

65

BREAKING

Cross-process DurableAgent.observe(runId) now throws `DURABLE_AGENT_OBSERVE_AGENT_MISMATCH` when the persisted run belongs to an agent with a different ID; the observing process must use the same agent ID as the process that started the run.

– Exact error code and constraint, no code example `@mastra/core@1.79.0`

13 **Microsoft Teams channel and per-resource connect routes** NEW

60

Adds Microsoft Teams as a supported channel in `@mastra/connect`, along with userConnectRoutes() for per-resource connection scoping so authenticated end users can connect their own accounts.

– Names channel and function, no example or config detail `@mastra/core@1.79.0`

THINNER COVERAGE BELOW

14 **Models panel for agent fallback management in Studio** NEW

55

Studio adds a Models panel showing an agent's primary model and fallback models in priority order. Users can reorder fallbacks, enable or disable them, and change the provider or model for static model lists, all at runtime.

– UI feature described without exact navigation path `product docs`

└──▷ BREAKING ON UPGRADE

! Agent controller sessions are now keyed by `sessionId`: controller.getSessionByResource(resourceId, scope) is removed, `scope` is removed from

createSession(), deleteSession() now takes `{ sessionId }`, and the session() / session.create() signatures in `client-js` are updated.

! Thread-row query contract changed: use `timeRange.traces` / `timeRange.threads` instead of `traces.timeRange`.

! querySpans() contract changed: `model` and `provider` filters are removed, and `cost`, `inputTruncated`, `outputTruncated`, `errorTruncated` are dropped from responses — requires updated storage adapters.

! Channel `resolveResourceId` and `resolveThreadId` contexts now require a `requestContext` field; code that builds these objects by hand (e.g. tests calling a resolver directly) must pass requestContext: new RequestContext().

! Cross-process DurableAgent.observe(runId) now throws `DURABLE_AGENT_OBSERVE_AGENT_MISMATCH` when the persisted run belongs to an agent with a different ID; the observing process must use the same agent ID as the process that started the run.

WAS THIS USEFUL? 👍 👎

Nous Research Hermes

Feedback ⓘ v0.22.3 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 consolidated its CLI, Bot mode and Desktop clients around a single shared gateway process with new drain-timeout and `hermes gateway ensure` lifecycle controls, added MCP tool-cost budgets and experimental webhook-driven MCP event subscriptions, moved its own MCP-server mode to a separate `mcp-serve` plugin, and shipped a global pause/resume control, worktree lifecycle management, and Krea 3 image generation.

└──▷ **WHAT SHIPPED · 24 FEATURES** most completely described first ? what's the number?

01 **Gateway restart and drain timeout controls** NEW 100

New config keys govern graceful shutdown: `agent.restart_drain_timeout` (overridable via `HERMES_RESTART_DRAIN_TIMEOUT`) sets how long in-flight chat turns may keep running when the gateway stops or restarts; `cron_drain_timeout` sets a floor drain timeout for cron jobs and `api_server` (`/v1`) runs; `restart_after_turn_timeout` makes `hermes gateway restart` (and `/restart`) wait for turns to finish before stopping. The generated `systemd` unit's `TimeoutStopSec` is auto-computed from the larger of `restart_drain_timeout` and `cron_drain_timeout` plus 10s, plus 30s headroom (minimum 60s); raising the drain timeout requires re-running `hermes gateway install` so systemd doesn't SIGKILL the gateway mid-drain.



Allow in-flight chat turns up to 60 s to finish gracefully when the gateway restarts, so users do not lose work mid-turn.

```yaml
restart_drain_timeout: 60
cron_drain_timeout: 120
restart_after_turn_timeout: 60
```

Override the drain timeout for a one-off gateway restart without editing config — useful in CI or maintenance scripts.

```
$ HERMES_RESTART_DRAIN_TIMEOUT=30 hermes gateway restart
```

— Full mechanism, exact keys, defaults, and migration step. product docs

02 **`hermes gateway ensure` discovery and startup command** NEW

95

`hermes gateway ensure [--json] [--timeout 60]` discovers or starts the active profile's session runtime without installing a service, emitting a JSON object with `state`, `reason_code`, and `endpoint` (exit 0 means ready); `--interrupt` skips the `y/N` prompt when active turns are running, and `--detach` starts the replacement gateway in the background when no gateway service is installed. Bot mode uses it to attach to or start the shared gateway before connecting over WebSocket at `/api/ws` with a one-use ticket, and the CLI uses it to locate or launch the profile's gateway before attaching.

Ensure the local gateway is running and attach to it — useful in a startup script or after a reboot before opening any Bots.

```
$ hermes gateway ensure --json
```

Check whether the local gateway is ready before kicking off automated scans — get a machine-readable status without risking a service install.

```
$ hermes gateway ensure --json --timeout 60
```

— Exact flags, JSON fields and runnable command given. product docs

03 **Worktree archiving, sweep and restore** NEW

95

Hermes automatically sweeps stale worktrees at startup and every 6 hours via the cron ticker, archiving dirty trees under `refs/hermes/archive/<tree>` and removing clean merged ones. `worktrees.archive_dirty: false` stops the sweep from ever deleting trees with uncommitted changes or unpushed commits, and `hermes`

`worktree restore <name>` (with optional `--path DIR`) recreates an archived worktree.

Switch the CLI to the new modern layout that never garbles on terminal resize, then lock worktree cleanup to read-only archiving so uncommitted work is never deleted.

```yaml
display.cli_layout: modern
worktrees.archive_dirty: false
```

Inspect accumulated worktrees after a long gateway-only run, then restore an archived one that had uncommitted changes.

```
$ hermes worktree restore my-feature-tree --path ~/restored/my-feature-tree
```

— Exact commands, config key and archive path given. product docs

04 **Per-tool spend limits in MCP config** NEW 90

MCP server config entries can declare `tool_costs`, `spend_limit`, and `cost_unit` keys so each tool call declares a credit cost against a per-chat budget ceiling; costed calls surface the cost in the approval prompt and track a running total per server per chat. A local `spend_limit: 0` entry overrides a portable plugin package's declared limit, and `tool_costs: {}` drops its declared costs, giving local config precedence over package defaults; malformed `tool_costs`, `spend_limit`, or `cost_unit` values cause the server to be skipped with a named-key error.

Cap how many credits an image-generation MCP server can spend in a single chat, so runaway tool calls don't exhaust your quota silently.

```yaml
trust: untrusted
tool_costs:
  create_image: 20
  animate:
    min: 5
    max: 20
    note: 'per animation'
  get_status: 0
spend_limit: 100
cost_unit: credits
```

— Full mechanism, override behavior and example config given. product docs

05 **MCP event subscriptions via webhook** NEW 90

Setting `events: true` on an MCP server config registers experimental `list_event_types`, `subscribe_event`, and `unsubscribe_event` utility tools, triggering agent runs through the gateway's webhook listener when upstream events arrive. Each subscription is stored as a dynamic webhook route (visible via `hermes webhook list`) with its own `whsec_` signing secret, with grant and replay cursor persisted in `mcp_event_subscriptions.json`; the per-route rate limit `platforms.webhook.extra.rate_limit` (default 30 per minute) applies to deliveries and can be raised for busy event types.

Subscribe Hermes to upstream MCP server events (e.g. new emails or incidents) so each arriving event starts an unattended agent run via the webhook listener.

yaml

```yaml
"my-event-server":
  events: true
  platforms:
    webhook:
      extra:
        public_url: "https://hermes.example.com"
```

— Named config keys, storage file and rate limit, with example. product docs

06 **Automatic backgrounding and process-wait notifications** NEW 90

Trailing-`&` shell commands (e.g. `python3 -m http.server 8765 &`) are now automatically run with `background=true` and tagged with an `auto_backgrounded` note, while ambiguous shapes like `server & curl ...`, `nohup` / `disown` / `setsid` wrappers, and command substitutions are refused with guidance. When process(action="wait") is called twice in a row on a silent running process, Hermes auto-enables `notify_on_complete` and delivers the exit code and final output as a later message instead of polling; the `process_wait_auto_notify` config key (set `false` to keep plain polling) and `display.background_process_notifications: off` profile setting disable this.

Disable the auto-notify behaviour globally so all process waits stay plain polling — useful in cron jobs or stateless pipelines where a later completion message cannot be received.

yaml

```yaml
process_wait_auto_notify: false
```

Turn off background-process completion notifications for a specific profile so the agent is never resumed by a background exit event.

```yaml
display.background_process_notifications: off
```

— Exact trigger conditions, config keys and disable options given. product docs

07 **CLI redesigned as a gateway client** BREAKING 90

The classic CLI is now a gateway client: closing the terminal detaches rather than cancelling the turn, with accepted work continuing at the gateway. Options `--image`, `--worktree`, `--no-restore-cwd`, `--run-budget`, `-v`, `--compact`, and bare `--create-if-missing` (without `-c <name>`) are now refused up front with exit code 2 and never fall back to a local agent; a `--safe-mode` or `--ignore-user-config` launch no longer reads a profile default model and must explicitly name one (e.g. `--provider openrouter --model anthropic/claude-sonnet-4`); slash commands outside the gateway CLI surface print `Unsupported gateway CLI command; use /help.` instead of executing.

— Lists every refused flag and exact error/migration behavior. product docs

08 **Bulk plugin updates with `--yes`** NEW 88

`hermes plugins update` with no name lists all updatable plugins and prompts `Update all N? [Y/n]`, updating each without one failure stopping the rest. `--yes` applies all available updates non-interactively, skipping consent prompts and keeping new capabilities ungranted when there's no terminal.

Apply all pending plugin updates unattended in a CI pipeline where there is no terminal to respond to prompts.

```
$ hermes plugins update --yes
```

— Exact command, flag and runnable example. product docs

09 **Hermes-as-MCP-server moved to mcp-serve plugin** BREAKING 85

Running Hermes as an MCP server is now a plugin, `hermes plugins mcp-serve` (also `hermes plugins install mcp-serve` / `hermes mcp-serve`), for clients such as Claude Code, Claude Desktop, Cursor, and Codex; it replaces the former built-in `mcp serve` subcommand, and `hermes mcp serve` now just prints the new spelling. MCP clients configured with `'args': ['mcp', 'serve']` must be updated to `['mcp-serve']`.

Install the mcp-serve plugin so Claude Desktop or Cursor can call Hermes tools and read messaging conversations over the MCP protocol.

```
$ hermes plugins mcp-serve
```

— Exact migration command and breaking-change note given. product docs

10 **`-c <title> --create-if-missing` session flag** NEW

80

The `-c <title> --create-if-missing` flag combination creates a titled session when none exists and resumes it when one does, enabling deterministic threads for programmatic callers.

Resume a named session from a bot or script, creating it on first run so every call lands in the same deterministic thread.

```
$ hermes -c "audit-bot" --create-if-missing -q "run nightly checks"
```

— Exact flags with a runnable example. product docs

11 **Gateway CLI slash command surface** NEW

80

The supported slash commands in gateway CLI mode now include `/discard <admission_id>` (acknowledge an unknown queued turn), `/approve <id> <choice>`, `/answer <id> <text>`, `/branch [title]`, `/model <model> [--provider name]`, `/compress [here [N] | <focus>] [--preview]`, and `/yolo [on|off]`.

— Enumerates every slash command verbatim. product docs

12 **Shared gateway process for all local Bots** IMPROVED

78

Bot mode now runs a single shared `hermes gateway` process per host that owns all local Bot profiles, eliminating per-Bot backend pools and any concurrency limit on warm Bots. `gateway.standalone: true` profile metadata keeps a dedicated gateway per profile instead, with Desktop attaching to it the same way.

Cleanly stop the shared gateway after a maintenance window or before a manual update.

```
$ hermes gateway stop
```

Pin a specific profile to its own dedicated gateway so it is never co-tenanted with other local profiles.

```yaml
gateway.standalone: true
```

— Named config key and architecture change, example commands given. product docs

13 **Explicit custom provider endpoint flag** NEW 78

`--provider custom --base-url <url> --api-key <key>` configures a fully explicit OpenAI-compatible endpoint; the API key is held in gateway memory for the session only and never written to disk.

— Exact flags and security note but no worked example. product docs

14 **Local/TUI session env passthrough and prefill messages** NEW 76

`client_env_passthrough` names extra non-secret toolchain variables (e.g. `GEM_HOME`) that a local CLI/TUI session's tools take from its launching shell. `prefill_messages_file`, a new profile setting, is resolved at session-creation time from the profile for local/TUI sessions served by the gateway, with edits applying to the next new session without a gateway restart.

Pass build-tool env vars from the launching shell into a local CLI/TUI session so tools like Bundler and pip resolve paths correctly.

```yaml
client_env_passthrough:
  - GEM_HOME
  - VIRTUAL_ENV
  - GOPATH
```

— Two named keys with mechanism, one example. product docs

15 **Session retention, spillover and image-replay limits** NEW 73

`sessions.retention_days` (default 90 days, minimum 24 hours) controls how long session spill files and transcripts are kept by the gateway's hourly housekeeping, while a generic per-result spillover threshold of 100,000 characters (scaled down automatically for small-context models) decides when results spill to file, with spill files retained as long as any session transcript references them. `vision.replay_recent_images` additionally controls whether the newest user images are re-sent on turns rebuilt from the session database.

— Named keys and defaults but no runnable example. product docs

16 **Global pause/resume for all turn types** NEW 65

A global emergency stop freezes all turn types at once — built-in ticker, managed-cron, misfire catch-up, kanban dispatch, and gateway turns including interactive CLI, TUI, and Desktop — showing `Hermes is paused` with the reason on each surface; in-flight turns finish and are never killed.

— Names affected surfaces and behavior but no command to trigger it shown. product docs

17 **Modern scrollback-safe CLI layout** NEW 62

`display.cli_layout` (values `modern` or `classic`) switches between a new scrollback-safe single-glyph row layout and the legacy boxed-panel layout.

Switch the CLI to the new modern layout that never garbles on terminal resize, then lock worktree cleanup to read-only archiving so uncommitted work is never deleted.

```yaml
display.cli_layout: modern
worktrees.archive_dirty: false
```

— Named config key and example but limited explanation of the mechanism. product docs

18 **Sandbox bind-mount privilege stripping** IMPROVED 60

Hermes strips `setuid` / `setgid` bits and file capabilities from sandbox bind-mount files (`TERMINAL_SANDBOX_DIR`) whenever a long-lived container detaches, reattaches, or is swept by the stale-container pass.

— Names the mechanism and trigger points but no config/command. product docs

THINNER COVERAGE BELOW

19 **Multi-group-chat membership for Bots** NEW 55

A New Group Chat picker checks how the active gateway runs group rooms before offering members, ensuring every selected Bot is reachable by that gateway. Bot group-chat membership is stored in backend-synced profile metadata so group assignments follow the profile across desktops, with backward compatibility for legacy single-group profiles.

— Describes behavior and storage but no exact UI path or command. product docs

20 **Plugin catalog additions and version bumps** NEW 55

Adds `provenance-gate`, `safeailabs-provider`, `burnclock`, `model-picker`, `hermes-bot-hq`, `cursor-acp`, `wireguard`, `empiriolabs`, `post-cortex`, `orquehelx`, `Promoat`, `hermes-board`, and `degeneracy-guard`, and `Requesty` to the plugin catalog, and bumps the `hyatlas` catalog pin to 5.0.0 and `hermes-teammates` to 1.1.0.

— Names every plugin and version bump, no usage detail. v0.22.3

21 **`replay_gap` field in turn events** NEW 52

Turn events now include a `replay_gap: true` field to indicate the client re-attached after falling behind the gateway's stream, while `here` still contains the complete reply.

— Names the field but no usage walkthrough. product docs

22 **Krea 3 image generation and editing** NEW 52

Krea 3 image generation and editing is available through the managed gateway with thinking and resolution controls, and is now the default image model for paid subscribers.

— Names the model and default status but no command or UI path. v0.22.3

23 **Customizable CLI status bar fields** NEW 50

`display.status_bar.fields` customizes which fields appear on the left side of the CLI status bar/line.

— Names the key but doesn't list available field values. product docs

24 **Microsoft Store install metrics slicing** NEW 45

`install_method: microsoft-store` opts Store installs into shared metrics so they can be sliced from other install types in observability dashboards.

— Names the key but no dashboard or setup detail. v0.22.3

└──▷ BREAKING ON UPGRADE

! The built-in `mcp serve` subcommand is replaced by the `mcp-serve` plugin; MCP clients configured with `'args': ['mcp', 'serve']` must be updated to `['mcp-serve']`.

! After raising `restart_drain_timeout`, you must re-run `hermes gateway install` so the generated `systemd` unit picks up the new `TimeoutStopSec` budget — otherwise systemd SIGKILLs the gateway mid-drain.

! `hermes mcp serve` no longer runs Hermes as an MCP server directly; users must install and use the `mcp-serve` plugin (`hermes plugins install mcp-serve` / `hermes mcp-serve`) instead.

! Options `--image`, `--worktree`, `--no-restore-cwd`, `--run-budget`, `-v`, `--compact`, and bare `--create-if-missing` (without `-c <name>`) are now refused up front with exit code 2 when the CLI runs against the gateway; nothing falls back to a local agent.

! A `--safe-mode` or `--ignore-user-config` launch no longer reads a profile default model and must explicitly name one (e.g. `--provider openrouter --model anthropic/claude-sonnet-4`).

! Slash commands not in the gateway CLI slash surface now print `Unsupported gateway CLI command; use /help.` instead of executing.

WAS THIS USEFUL?  

Agno (formerly Phidata)

Feedback ⓘ v3.1.8 NOTES ↗

RANK

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

> Agno v3.1.8 adds a PostgreSQL-backed checkpoint store for multi-replica sandbox deployments and configurable checkpoint retention, alongside new Company Routine executor support, stricter AgentOS startup validation, and a breaking change to Claude's default model.

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

01 PostgresCheckpointStore for sandbox checkpoints NEW

80

Adds `PostgresCheckpointStore` for storing sandbox checkpoints directly in PostgreSQL, enabling multi-replica deployments to restore sessions without S3 or shared filesystems. Checkpoints are chunked, written transactionally, deduplicated by SHA-256, and verified on read.

— Names class and storage mechanism but no setup example given v3.1.8

02 Configurable checkpoint retention via retain_checkpoints

80

BREAKING

Adds `retain_checkpoints=None` config option to preserve every sandbox checkpoint; default behavior now retains only the three newest checkpoints per session. Opening a checkpoint for an older run can raise `CheckpointObjectNotFound` unless `retain_checkpoints=None` is set to restore unlimited-retention behavior.

— Exact config flag and exception name, migration path given v3.1.8

03 Claude default model and parameter handling change BREAKING

65

Claude() now defaults to `claude-sonnet-5-5`; non-default `temperature`, `top_p`, and `top_k` values are silently dropped with a warning for newer Claude models across Anthropic, Bedrock, Foundry, and Vertex providers.

— Names exact model and dropped params, no migration code v3.1.8

04 Startup validation for durable-queue checkpoint stores

60

BREAKING

AgentOS now fails at startup when a durable-queue sandbox uses the host-local default checkpoint store, surfacing the misconfiguration immediately rather than on first submission; a shared store such as `PostgresCheckpointStore` or `S3CheckpointStore` must be configured instead.

— Explains behavior change but no exact config key shown v3.1.8

THINNER COVERAGE BELOW

05 **Framework managers as Company Routine executors** NEW

45

Framework managers (the internal Product Manager or OS Manager) can now act as Company Routine executors via declarations, the Routine API, and human-chat configuration, with an optional weekly Product Manager review.

— Names API and roles but no concrete usage shown v3.1.8

└──▷ **BREAKING ON UPGRADE**

! Only the three newest sandbox checkpoints are retained by default; opening a checkpoint for an older run can now raise `CheckpointObjectNotFound`. Set `retain_checkpoints=None` to restore unlimited-retention behavior.

! AgentOS raises an error at startup if a durable-queue sandbox is configured with the host-local default checkpoint store — a configuration that previously only failed on first submission. You must configure a shared store such as `PostgresCheckpointStore` or `S3CheckpointStore`.

! Claude() now defaults to `claude-sonnet-5-5`; non-default `temperature`, `top_p`, and `top_k` values are silently dropped with a warning for newer Claude models across Anthropic, Bedrock, Foundry, and Vertex providers.

WAS THIS USEFUL? 👍 👎

Mads Lorentzen AI Job Search

Feedback ⓘ v1.7.3

NOTES ↗ RANK

The job search that runs on your machine. AI job application framework built on Claude Code: evaluate postings, tailor CVs, write cover letters, prep interviews.

> AI Job Search shipped a major security hardening pass — a prompt-injection sanitizer for job postings plus least-privilege permission enforcement across agent definitions and settings — alongside a one-command portal CLI installer, new PDF layout verification, host validation for portal and ATS lookups, and several workflow command upgrades.

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

01 Least-privilege agent permissions and settings guardrails

95

BREAKING

`.claude/settings.json` now requires `ask` and `deny` blocks that hold in every permission mode, blocking uploads via `curl`, `wget`, `scp`, `ssh`, and `nc` and reads of `.env` and credential folders; forks with customised settings must add these blocks verbatim. Every agent definition in `.claude/agents/*.md` must declare a `tools:` list limiting `posting-fetcher`, `posting-scorer`, and `application-reviewer` to read, fetch, and search operations, and skill frontmatter in `SKILL.md` may grant only tools already allowed in `settings.json`, validated against an exact allowlist in `tools/security_guards.py`. The bare `Bash` grant no longer pre-approves `lualatex` / `xelatex` or `rm -f` build-file cleanup, so these now prompt in default permission mode, and `SECURITY.md` documents every shipped grant, its lifetime, and its guard.

— Full mechanism, exact config keys, and migration notes included. v1.7.3

02 Prompt-injection sanitizer for job postings NEW

78

`tools/posting_guard.py` sanitizes every job posting before `/apply`, `/rank`, or `/interview` reads it — stripping hidden HTML, zero-width and bidi characters, quoting instruction-shaped lines, and flagging mixed-script words and URLs found only in hidden content.

— Mechanism and scope fully named; no manual invocation given. v1.7.3

03 PDF layout measurement tool NEW

75

`tools/verify_layout.py` measures per-page text extent, bottom whitespace, largest internal gap, footer collisions, and stranded entry headers at page breaks, catching layout failures that `verify_pdf.py` missed.

— Mechanism and scope named; no usage example given. v1.7.2

04 **Host Validation Guards in Portal CLIs and /apply** NEW

73

Adds host validation guards before any network or drafting action: `jobindex-search detail` and `linkedin-search detail` now parse the real host from the posting URL and exit 1 with the `BAD_ID` contract before any network request when the host does not match, while `/apply` Step 1 verifies the posting host against installed portals and six standard ATS apexes, flagging look-alikes and unknown hosts before any drafting tokens are spent.

— Exact commands and error contract named. Command step and scope named; exact apex list omitted.

v1.7.2

05 **Pure-function job key generation** NEW

67

`tools/job_key.py` owns `seen_jobs.json` key generation as a pure function of the posting, with an `--audit` flag to report legacy entries.

— File, config target, and flag named. v1.7.2

06 **403-retry page fetcher** NEW

65

`tools/fetch_page.py` acts as the 403-retry fetcher — it performs a single GET, checks `robots.txt` first, and has no upload path.

— Mechanism and file named; no usage example given. v1.7.3

07 **Methodology file version bumps flagged by upstream checker**

65

BREAKING

Three methodology files moved to new versions — `job-application-assistant/SKILL.md` 1.3.4 -> 1.3.6, `09-web-research.md` 1.1.1 -> 1.1.5, and `05-cv-templates.md` 1.4.5 -> 1.4.7 — which `check_upstream_updates.py` will flag, requiring fork maintainers to re-read them before merging local edits.

— Exact version numbers given; migration step named. v1.7.3

08 **Batch stale outcome resolution** NEW

65

`/outcome stale [N]` batch-resolves applications quiet for 60+ days with explicit confirmation.

— Exact subcommand syntax and threshold given. v1.7.2

09 **Independent Projects profile section sourcing** NEW

62

`/setup` now ingests project writeups from `documents/projects/`, and `/expand` mines public GitHub repositories, both populating the `## Independent Projects` profile section.

— Named commands, path, and section header. v1.7.2

10 **One-command portal CLI installer** NEW 60

`python3 tools/install_portal_clis.py` installs every portal CLI with one cross-platform command.

— Runnable command given; little detail on what gets installed. v1.7.3

11 **Deadline retirement and portal CLI fallback in /rank** IMPROVED 60

`/rank` now retires scored jobs whose deadline has passed even when no new postings arrived, and tries an installed portal CLI's `detail` subcommand before falling back to `WebFetch` for portal URLs.

— Behavior change described; no command to invoke directly. v1.7.3

12 **Profile preservation in /reset** IMPROVED 60

`/reset` now preserves profile headings, the Languages table header, and `framework_version` frontmatter while clearing nested document entries and keeping folder scaffolds.

— Clear before/after behavior; no new flag surfaced. v1.7.3

THINNER COVERAGE BELOW

13 **Lighter endpoint and User-Agent for freehire-search** IMPROVED 45

`freehire-search` uses a lighter endpoint for `--no-description` and identifies this project in its User-Agent.

— Named flag and change, but scope stays thin. v1.7.3

14 **ASCII dates flag for PDF verification** NEW 43

Adds `--ascii-dates` flag to `verify_pdf.py`.

— Bare flag name; no explanation of what it changes. v1.7.3

15 **Template intermediate retention in /add-template** IMPROVED 36

`/add-template` now retains a registered template's intermediates in `build/`.

— Bare behavior change with minimal detail. v1.7.2

└──▷ BREAKING ON UPGRADE

! `.claude/settings.json` now requires `ask` and `deny` blocks; the security guard will fail on any fork that omits them. Customised `settings.json` files must add these two blocks verbatim while keeping their own `allow` lines.

! Any `.claude/agents/*.md` file without a `tools:` list will fail guard check 5. The shipped `gemini-research-expert.md` has been removed; forks that used it must restore it from history and add a `tools:` line.

! The bare `Bash` grant no longer pre-approves `lualatex` / `xelatex` or `rm -f` build-file cleanup; these will now prompt in default permission mode.

! Three methodology files moved to new versions (`job-application-assistant/SKILL.md` 1.3.4 -> 1.3.6, `09-web-research.md` 1.1.1 -> 1.1.5, `05-cv-templates.md` 1.4.5 -> 1.4.7); `check_upstream_updates.py` will flag them and fork maintainers must re-read them before merging local edits.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

Composio

Feedback ⓘ @composio/vercel@0.13.0 NOTES ↗ RANK

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

> Composio's October releases focus on JSON Schema fidelity across its AI SDK providers (Vercel, Claude Agent SDK, LlamaIndex) and strict-mode schema fixes for OpenAI/Anthropic, alongside a broad credential-handling and client-configuration redesign in the core/slim SDK that adds organization API key propagation and toolkit deprecation warnings.

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

01 Full JSON Schema fidelity across AI SDK providers IMPROVED

90

In the Vercel provider, tool JSON Schema is now passed to the AI SDK via jsonSchema(), exposing keywords Zod previously dropped: `format: "binary"`, `contentEncoding`, `contentMediaType`, `title`, `default`, and `examples`. The Vercel, LlamaIndex, and Claude Agent SDK providers now expose each tool's own JSON Schema to the model, and `@composio/claude-agent-sdk` builds tool input schemas with the new Zod v4 builder from `@composio/json-schema-to-zod/v4`, preserving `format`, `contentEncoding`, `contentMediaType`, and `type: 'integer'` (e.g. `{ type: 'string', format: 'binary' }` no longer drops to `{ type: 'string' }`). That builder ships as a new `@composio/json-schema-to-zod/v4` entry point exposing `jsonSchemaToZod` and `jsonSchemaToZodShape` for Zod v4 (requires zod `>= 4.1.13`), forwarding those same keywords into schema metadata so z.toJSONSchema() can round-trip them, with a `withoutAnnotations: true` option to suppress them; the root entry point still builds Zod v3 schemas.

Build a Zod v4 schema from a JSON Schema so that `format`, `contentEncoding`, and `contentMediaType` survive a round-trip through z.toJSONSchema().

```typescript
import { jsonSchemaToZod } from '@composio/json-schema-to-zod/v4';
import { z } from 'zod/v4';

const schema = jsonSchemaToZod({
  type: 'object',
  properties: {
    avatar: { type: 'string', contentEncoding: 'base64', contentMediaType: 'image/png' },
    created: { type: 'string', format: 'date-time' }
  }
});

console.log(z.toJSONSchema(schema));
```

Suppress metadata annotations on the Zod v4 schema when you only need runtime validation and do not need z.toJSONSchema() to re-emit them.

```typescript
import { jsonSchemaToZod } from '@composio/json-schema-to-zod/v4';

const schema = jsonSchemaToZod(
  { type: 'object', properties: { id: { type: 'integer' } } },
  { withoutAnnotations: true }
);
```

— Names every preserved keyword and gives runnable code examples. `@composio/vercel@0.13.0` `@composio/claude-agent-sdk@0.13.0` `@composio/json-schema-to-zod@0.4.0` `@composio/cli@0.5.1-beta.488`

02

Credential and client-config API redesign in SDK

BREAKING

90

Credential and scope entries from `defaultHeaders` or createSession({ headers }) now fold into SDK options: `x-user-api-key` or `x-org-api-key` becomes the per-request credential, `x-api-key` becomes the project key, and `x-org-id` / `x-project-id` become `orgId` / `projectId`; setting more than one of `x-api-key`, `x-user-api-key`, and `x-org-api-key` simultaneously now throws a `ComposioError` instead of failing silently at the API. getConfig() now reports the folded values (e.g. `apiKey: null`, `userApiKey`) instead of listing them under `defaultHeaders`, and the `COMPOSIO_CUSTOM_HEADERS` environment variable is replaced by the `defaultHeaders` option (`COMPOSIO_ALLOW_INSECURE_HTTP=1` is

still honored). On the client from getClient(), `apiKey`, `userApiKey`, `orgApiKey`, and `logger` are removed or made read-only, as are `baseURL`, `maxRetries`, `timeout`, and `logLevel` — use getConfig() or the client's `authMode` / `scope` for credentials, and withOptions() to derive a modified client. The deprecated `customConnectionData` now requires `val.access_token` for `S2S_OAUTH2` connections, and `@composio/core` now requires `@composio/client` `2.0.0-rc.19` (up from `rc.18`).

— Every renamed field, error, and env var named with migration path.

`@composio/slim@0.26.0` `@composio/core@0.26.0`

---

03 **`tool.inputSchema` now an AI SDK Schema type** BREAKING

85

`tool.inputSchema` is now an AI SDK Schema type instead of a Zod schema; code calling inputSchema.safeParse() directly will break and must migrate to asSchema(tool.inputSchema).validate(value) from `ai`.

— Exact migration call given, no usage example. `@composio/vercel@0.13.0`

---

04 **Strict-mode JSON Schema fixes for OpenAI and Anthropic**

IMPROVED

85

`toStrictJsonSchema` now strips unsupported string `format` values from strict tool schemas, keeping only `date-time`, `time`, `date`, `duration`, `email`, `hostname`, `ipv4`, `ipv6`, and `uuid` — previously a single tool with `format: 'uri'` or `format: 'path'` caused every strict OpenAI request to fail with a 400. It also folds OpenAPI `nullable: true` into the type array (e.g. `['object', 'null']`) instead of forwarding the `nullable` keyword, preventing Anthropic strict tool-use 400 errors on object schemas. In the Vercel provider, the model now sees this strict-rewrite output directly, fixing failures caused by Zod injecting a `pattern` alongside `format: "email"` that structured outputs rejected.

— Concrete before/after failure modes and allowlist named. `@composio/vercel@0.13.0` `@composio/core@0.26.0`

---

05 **Organization API key credential propagation** IMPROVED

85

With `apiKey: null` and only an organization API key supplied (via the `orgApiKey` option, `COMPOSIO_ORG_API_KEY`, or an `x-org-api-key` default header), the `x-org-api-key` credential now propagates to every request, trigger-channel authorization, and `session.mcp.headers` — previously only organization-owner operations carried the credential.

— Names option, env var, header, and before/after behaviour. `@composio/slim@0.26.0`

---

06 **Deprecation warnings for toolkit versions** NEW

78

The SDK now warns once per toolkit version when the API signals deprecation via the `Composio-Deprecated-Toolkits` response header, including the version, sunset date, and pinning instructions; suppress with `logLevel: 'error'` or `COMPOSIO_LOG_LEVEL=error`.

Silence deprecation warnings for toolkit versions flagged by the `Composio-Deprecated-Toolkits` header in CI pipelines where noise must be suppressed.

```
$ COMPOSIO_LOG_LEVEL=error node your-agent.js
```

— Exact header name and suppression command given. `@composio/slim@0.26.0`

THINNER COVERAGE BELOW

07 Base64 content-encoding validation restricted IMPROVED

55

`contentEncoding: 'base64'` validation is now restricted to parameters explicitly marked with that encoding; `format: 'binary'` and other encodings such as `base16` are now treated as plain strings.

— States mechanism but no usage example. `@composio/slim@0.26.0`

└──▷ BREAKING ON UPGRADE

- ! `tool.inputSchema` is now an AI SDK `Schema` instead of a Zod schema — code calling inputSchema.safeParse() directly will break and must migrate to asSchema(tool.inputSchema).validate(value) from `ai`.
- ! The `zod` peer dependency is now `^4.1.13` (was `^4.0.0`); the v4 builder throws at load time on an older zod — upgrade zod to 4.1.13 or newer before upgrading `@composio/claude-agent-sdk`.
- ! The `COMPOSIO_CUSTOM_HEADERS` environment variable is no longer read by the API client; pass custom headers via `defaultHeaders` instead.
- ! Setting more than one of `x-api-key`, `x-user-api-key`, and `x-org-api-key` now throws a `ComposioError` rather than letting the API reject the request silently.
- ! On the client returned by getClient(), `apiKey`, `userApiKey`, `orgApiKey`, and `logger` are removed or read-only; read credentials from getConfig() or `authMode` / `scope`, and use withOptions() to change `baseURL`, `maxRetries`, `timeout`, or `logLevel`.
- ! A per-call `headers` option on the API client can no longer name a credential or scope header (`x-api-key`, `x-user-api-key`, `x-org-api-key`, `x-org-id`, `x-project-id`).
- ! getConfig() no longer lists folded credential/scope entries under `defaultHeaders`; those values appear instead as `apiKey`, `userApiKey`, `orgApiKey`, `orgId`, or `projectId`.
- ! The deprecated `customConnectionData` now requires `val.access_token` for `S2S_OAUTH2` connections.
- ! `COMPOSIO_CUSTOM_HEADERS` is no longer read by the API client; pass custom headers via `defaultHeaders` instead.

! `apiKey`, `userApiKey`, `orgApiKey`, and `logger` are removed from the object returned by getClient(); read keys from getConfig() and use `authMode` / `scope` on the client.

! `baseURL`, `maxRetries`, `timeout`, and `logLevel` on the client returned by getClient() are now read-only; use withOptions() to produce a modified client.

! Setting more than one of `x-api-key`, `x-user-api-key`, and `x-org-api-key` now throws a `ComposioError` at SDK level.

! The deprecated `customConnectionData` now requires `val.access_token` for `S2S_OAUTH2`.

! Requires `@composio/client` `2.0.0-rc.19` (up from `2.0.0-rc.18`).

WAS THIS USEFUL? 👍 👎

D4Vinci Scrapling

Feedback ⓘ v0.5 NOTES ↗ RANK

A Web Scraping framework that handles everything from a single request to a full-scale crawl! Don't be shy, join here: https://discord.gg/EMgGbDceNQ and follow here for daily tips and tricks: https://x.com/Scrapling_dev

> Scrapling v0.5 adds network traffic recording, browser session state persistence, Shadow DOM extraction, configurable crawl scheduling, and a much expanded MCP server for AI agents — alongside breaking changes across fetching, parsing, spiders, and browser automation.

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

01 **Network request recording in StealthySession** NEW 88

Setting `record_requests=True` on `StealthySession` records completed network traffic across page loads, actions, and tabs. Traffic can be filtered by URL, method, resource type, or status via session.network.search(url_pattern=..., limit=...), or a single response retrieved by ID; the default limit is 1,000 requests, and it works in both sync and async sessions.

— Names exact flag and search API with defaults v0.5

02 **Shadow DOM piercing for extraction** NEW 79

Adds `pierce_shadow=True` to browser fetchers and sessions, and `--pierce-shadow` to the CLI, to extract open Shadow DOM content and query it with CSS or XPath without modifying the live page.

— Names flag, CLI option, and query mechanism v0.5

03 **Browser session state persistence** NEW 72

New save_state(path) and load_state(path) methods on browser sessions persist and restore cookies, local storage, and IndexedDB, stored in Playwright-compatible JSON.

— Exact method names and storage format given v0.5

04 **Expanded MCP server for AI agents** IMPROVED 67

The MCP server now supports complete browser workflows for AI agents — forms, clicks, hovers, scrolls, keyboard input, and dialogs — plus page extraction as HTML/Markdown/text, custom JavaScript execution including async, PNG/JPEG screenshots of elements or full page, network activity inspection, and Shadow DOM interaction.

— Lists capabilities but no concrete invocation shown v0.5

05 **Configurable crawl strategies for CrawlSpider** NEW

64

Adds `crawl_strategy='bfs'` (oldest-first) and `crawl_strategy='dfs'` (newest-first) scheduling options to `CrawlSpider` and its subclasses including `ContentSpider`.

— Names exact option and affected classes v0.5

THINNER COVERAGE BELOW

06 **Breaking changes across core components** BREAKING 35

v0.5 introduces breaking changes across fetching, parsing, spiders, and browser automation; a Migrating to v0.5 guide is provided at the docs site before upgrading.

— Points to migration guide but no specifics listed v0.5

└──▷ **BREAKING ON UPGRADE**

! This release introduces breaking changes across fetching, parsing, spiders, and browser automation — consult the Migrating to v0.5 guide at `https://scrapling.readthedocs.io/en/latest/migrating-to-v0.5.html` before upgrading.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Snowflake Cortex AI

Feedback ⓘ

Oct 23, 2026: Retrieve Cortex Search (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 added a SQL table function for running Cortex Search queries directly within SQL statements.

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

01 **CORTEX_SEARCH_RETRIEVE table function** NEW 69

The new `CORTEX_SEARCH_RETRIEVE` table function (Public Preview) runs low-latency Cortex Search queries directly from SQL, reusing the service's existing serving resources, and its results can be combined with standard SQL joins and aggregations.

— Names exact function and capabilities but no usage example given.

Oct 23, 2026: Retrieve Cortex Search (P…

NEW INTEGRATION MCP server catalog integrations for Atlassian, GitHub, Google Workspace, Salesforce, Slack, and Snowflake

NEW INTEGRATION AI client support for CoCo Desktop, CoCo CLI, Claude, and Cursor via MCP

Oct 19, 2026: Tools for Cortex AI Gateway (Preview) NOTES ↗ STABLE

Tools for Cortex AI Gateway (Preview) lets ACCOUNTADMIN-role admins manage MCP servers, control tool exposure, and audit agent activity via `AGENT_TRACE_TABLE`.

Launches Tools for Cortex AI Gateway (Preview), connecting AI clients to business tools via the Model Context Protocol (MCP).

Adds `AGENT_TRACE_TABLE` table function to review tool-call activity for MCP servers within a Snowflake account.

Grants access to MCP servers for end users by assigning the `SNOWFLAKE$OPS.MCP$USER` database role to account roles, enabling automatic discovery in CoCo Desktop and CoCo CLI.

Supports adding custom MCP servers not in the catalog by providing a URL, enabling internal tools and custom integrations.

Allows administrators (requiring the `ACCOUNTADMIN` role) to configure which tools each MCP server exposes — for example, permitting read-only tools while blocking

> write and delete tools.
>
> +1 more

WAS THIS USEFUL?  

Hugging Face

Feedback ⓘ Passkeys for sign-in and MFA NOTES ↗

RANK

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

> Hugging Face shipped an OpenAI-compatible REST endpoint with drop-in Python/JS client support for its inference router, plus passkey-based passwordless sign-in and MFA.

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

01 OpenAI-compatible inference router endpoint and clients NEW

88

Hugging Face added `https://router.huggingface.co/v1/chat/completions` as an OpenAI-compatible REST endpoint, letting any HTTP client reach hundreds of open-weights models by swapping a single URL. It supports drop-in replacement for both the OpenAI Python client and the OpenAI JavaScript client by setting `base_url` to the HF router with no other code changes, and InferenceClient.chat_completion() gained an explicit `provider` parameter accepting a named provider string or `'auto'` for automatic selection.

Route chat completions through the HF provider network using an existing OpenAI Python client — no SDK change needed.

python

```python
client = OpenAI(
    base_url="https://router.huggingface.co/v1",
    api_key=os.environ["HF_TOKEN"],
)
```

Pin inference to a specific backend provider using InferenceClient to ensure reproducible routing in production pipelines.

python

```python
result = client.chat_completion(
    provider="<specific-provider>",
    messages=[{"role": "user", "content": "Hello"}],
)
```

— Names exact endpoint, params and shows runnable code product docs

02 Passkeys for sign-in and MFA NEW

63

Hugging Face added passkey support (fingerprint, face, screen lock, or hardware security key) as a primary sign-in method replacing passwords, and as a second factor alongside an existing authenticator app. Enrollment happens via the Authentication settings page.

Enroll a passkey (e.g. Touch ID or a hardware security key) so you can sign in without a password or add it as a second factor.

> 📍 1. Sign in to Hugging Face and go to Settings › Authentication. 2. Under `Passkeys`, click `Add a passkey`. 3. Follow your device or browser prompt to register your fingerprint, face, screen lock, or security key. 4. Once saved, use the passkey at next sign-in in place of your password, or as a second factor alongside your authenticator app.

— Describes mechanism and exact UI path to enroll `Passkeys for sign-in and MFA`

WAS THIS USEFUL? 👍 👎

Hugging Face OpenEnv

Feedback ⓘ v0.10.0 NOTES ↗ RANK

An interface library for RL post training with environments.

> OpenEnv v0.10.0 adds policy enforcement for validation runs, execution-budget reporting, a new helium_browser_env playground, and several refinements to harbor_env session metadata and evaluation export/rubric behavior.

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

01 **New helium_browser_env environment** NEW 55

Adds `helium_browser_env`, a new environment shipped with a README, catalog card, Hugging Face Space deploy, and an interactive playground.

— Names env and gives concrete deploy/playground access points v0.10.0

02 **HarborSessionFactory metadata forwarding in harbor_env**

IMPROVED 55

`harbor_env` now forwards `purpose`, `provider`, and `eval_sampling` from `HarborSessionFactory`.

— Names exact fields and class but no usage example v0.10.0

03 **Public/no-network policy enforcement for validation** NEW 45

Validation runs now enforce and measure `public` and `no-network` policies.

— Names the two policies but no config surface or flag given v0.10.0

04 **Execution budget reporting in tbench2_env** IMPROVED 40

The reset observation in `tbench2_env` now includes execution budget reporting.

— Names env and observation field but no format/limit detail v0.10.0

05 **Container pre-hooks before rubric child evaluation** IMPROVED 30

Container pre-hooks now run before rubric children are evaluated.

— Describes ordering change but no config or command named v0.10.0

06 **TextArena game end shown in final observation** IMPROVED 25

The final observation now shows how a TextArena game ended.

— Bare one-line description with no mechanism or field name v0.10.0

07 **Independent conversations preserved in evaluation exports**

IMPROVED 20

Evaluation exports now preserve independent conversations rather than merging them.

— No named surface or mechanism beyond one-line description v0.10.0

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.42.0 NOTES ↗

RANK

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

> The OpenAI TypeScript SDK's two releases added small capabilities around voice management and streaming response metadata, including voice deletion, custom voice consent handling, and new usage/trigger fields in API response types.

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

01 **Cache-write usage and compaction trigger metadata** NEW

27

API response types now include 30-minute cache-write usage counts and compaction trigger IDs.

— Describes new fields but no field names or endpoints specified. v7.40.0

02 **Voice deletion and consent management** NEW

23

The API client gained support for deleting voices and managing consent for custom voices.

— No endpoint names or parameters given, just prose description. v7.40.0 v7.42.0

NEW INTEGRATION Vertex AI support for the Responses API

v7.41.0 NOTES ↗ STABLE

OpenAI TypeScript SDK v7.41.0 adds Vertex AI as a backend for the Responses API.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v7.41.0 https://github.com/openai/openai-node.git
# already have the repo? check out this version:
$ git checkout v7.41.0
```

WAS THIS USEFUL? 👍 👎

Tracer Cloud OpenSRE

Feedback ⓘ v0.1.2026.10.22 NOTES ↗

RANK

OpenSRE is an open-source toolkit for building AI SRE agents.

> OpenSRE's latest release adds an OTel-native payment watch with retained lifecycle reports, while retiring the built-in X Twitter MCP and Opsgenie integrations.

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

01 Retirement of X Twitter MCP and Opsgenie integrations

50

BREAKING

The built-in X Twitter MCP integration and the Opsgenie integration (along with its documentation) have been removed from OpenSRE; any workflows depending on either integration will stop working.

— Names both removed integrations but no migration path given v0.1.2026.10.22

02 OTel-native payment watch with lifecycle reports NEW

20

OpenSRE adds an OTel-native payment watch capability with retained lifecycle reports, as noted in the v0.1.2026.10.22 release summary; no further mechanism or configuration details are provided.

— Mentioned only in summary line, no elaboration or usage details v0.1.2026.10.22

└──▷ **BREAKING ON UPGRADE**

! The built-in X Twitter MCP integration has been removed; any workflows depending on it will stop working.

! The Opsgenie integration has been removed; any workflows depending on it will stop working.

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v2.1.0 NOTES ↗ RANK

The Anthropic Python SDK provides typed Python clients for Claude APIs, streaming responses, and tool use.

> Anthropic Python SDK v2.1.0 and v2.0.0 expanded user profile and spend limits API fields, while v2.0.0 also introduced a breaking API version upgrade.

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

01 **Enforcement fields on user profiles** NEW 45

User profile objects returned by the API now include `enforcement_status`, `actions`, and `filter` fields.

— Names exact fields but no behavior explained v2.1.0

02 **Spend limits API filtering and scope** NEW 45

The spend limits API gains a `workspace_id` filter and a `usage_tier` scope type.

— Names exact fields but no behavior explained v2.0.0

03 **API version 2026-10-21 upgrade** BREAKING 35

Adds support for API version `2026-10-21`; upgrading to this version is called out as a breaking change in this release.

— Breaking change noted but no migration detail given v2.0.0

└──▷ **BREAKING ON UPGRADE**

! Upgrading to API version `2026-10-21` is a breaking change introduced by this release.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.185 NOTES ↗ RANK

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

> HyperFrames shipped 4K transparent video rendering with opt-in capture motion blur, alongside a series of render-pipeline performance upgrades spanning Lambda parallelism, audio processing, large-file uploads, and a new echo-smear registry component.

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

01 4K transparent MOV/WebM render output NEW

80

Extends `cloud render --resolution 4k` to accept `--format mov` and `--format webm`, enabling 4K transparent video output that was previously refused by the CLI and render server.

Render a 4K transparent overlay (e.g. a motion graphic for compositing) as a WebM with an alpha channel.

```
$ hyperframes cloud render --resolution 4k --format webm
```

— Names exact flags and gives a runnable command v0.8.185

02 Capture motion blur via data-motion-blur NEW

80

Introduces `data-motion-blur` on the composition root as an opt-in capture motion blur feature, blurring fast-moving elements across sub-frame samples with configurable shutter angle and sample count; it respects device scale so 2x and 4K renders blur at full resolution.

Enable motion blur on a composition to smooth fast-moving elements across sub-frame samples during render.

```html
<div data-motion-blur>...</div>
```

— Mechanism and attribute both specified with usage example v0.8.185

03 Automatic hosted-asset upload for large media IMPROVED

60

`cloud render` now automatically uploads and caches media files of 5 MB or more as hosted assets before the project is zipped, removing the previous project upload-size ceiling; re-renders send only new or changed media.

— Behavior and threshold given but it's automatic, no user action v0.8.184

THINNER COVERAGE BELOW

04 **Large source file upload support** IMPROVED 55

Enables `lambda sites create` and Lambda renders to upload source files over 2 GiB and over 5 GB respectively.

— Names command and exact size limits lifted v0.8.183

05 **Parallel check execution across Chrome instances** IMPROVED 55

`check` now splits its samples across a second Chrome instance when a spare core and memory are available, reducing check time, and a repeat `check` from an agent reuses an already-running check process with the CLI loaded and Chrome warm.

— Names the command and mechanism but no flags to set v0.8.183

06 **Increased Lambda chunk capture parallelism** IMPROVED 50

AWS Lambda chunks now run as many capture browsers as the function can host — five instead of three on a 10 GB function — so heavy 4K chunks finish sooner.

— Concrete numbers given but infra-side, nothing to invoke v0.8.184

07 **Parallel audio pipeline for cloud and local renders** IMPROVED 50

Cloud renders now mix audio on their own Lambda in parallel with chunk rendering, build the audio track independently and decoupled from the main render pipeline, and read only the audio segment the edit uses from a file link instead of downloading the whole file; local renders now check audio peak in parallel segments, matching the cloud plan behavior.

— Mechanism detailed but automatic, no flag or command to invoke v0.8.183

08 **Multi-worker render parallelism improvements** IMPROVED 45

Multi-worker MP4 renders now start encoding while frames are still being captured, producing byte-identical output and cutting overall render time; separately, renders that fast capture refuses now keep their parallel workers instead of being pinned to one or three.

— Describes mechanism but no user-facing control to act on v0.8.183 v0.8.184

09 **echo-smear registry component** NEW 45

Adds `echo-smear` to the Registry: a streak of copies that trails an element as a GSAP timeline moves it faster.

— Names the component but gives no usage snippet v0.8.183

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.36.0 NOTES ↗

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

v3.36.0 adds 30-minute cache-write usage counts, compaction trigger IDs, custom voice consent management, and a native synchronous WebSocket option type.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v3.36.0 https://github.com/openai/openai-python.git
# already have the repo? check out this version:
$ git checkout v3.36.0
```

Adds 30-minute cache-write usage counts to API usage tracking.

Adds compaction trigger IDs to the API surface.

Adds custom voice consent management to the API.

Adds a native synchronous WebSocket option type via new type definitions.

NEW INTEGRATION Vertex AI support for the Responses API

WAS THIS USEFUL? 👍 👎

Anthropic TypeScript SDK

Feedback ⓘ sdk-v1.0.0

NOTES ↗ RANK

The Anthropic TypeScript SDK provides typed JavaScript and TypeScript clients for Claude APIs, streaming, and tool use.

> The Anthropic TypeScript SDK added new fields to user profile objects and expanded spend limit listing with workspace-level filtering and a new scope type.

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

01 **Spend limits listing filters** NEW 65

The list spend limits API call now accepts a `workspace_id` filter parameter, enabling per-workspace spend queries, and supports `usage_tier` as a new scope type when listing spend limits.

— Names exact parameter and scope type but no usage example sdk-v1.0.0

THINNER COVERAGE BELOW

02 **User profile object fields** NEW 28

User profile objects in the API gain `enforcement_status`, `actions`, and `filter` fields.

— Bare field names with no explanation of behavior or use sdk-v1.1.0

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

Crusoe Cloud

Feedback ⓘ snapshot-20261022 NOTES ↗ RANK

Crusoe Cloud provides GPU infrastructure and managed inference for AI training and deployment.

> Crusoe Cloud added Multi-LoRA support for Self-Serve Deployments and refreshed the DeepSeek-V4.1-Flash checkpoint.

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

01 Multi-LoRA for Self-Serve Deployments NEW

75

Deploy a base model once and hot-load multiple LoRA adapters at runtime, each exposed as its own model ID on the same OpenAI-compatible endpoint. This allows comparing multiple fine-tuned checkpoints side-by-side (e.g. during LoRA ablation or evaluation) without spinning up separate GPU deployments.

Compare multiple fine-tuned checkpoints side-by-side without spinning up separate GPU deployments — useful during LoRA ablation or evaluation runs.

> 📍 Deploy a base model with multiple LoRA adapters, each addressable by its own model ID on the same OpenAI-compatible endpoint.

— Explains mechanism and endpoint behavior with usage example snapshot-20261022

THINNER COVERAGE BELOW

02 DeepSeek-V4.1-Flash checkpoint update IMPROVED

40

`deepseek-ai/DeepSeek-V4.1-Flash` updated to the 1017 checkpoint with refreshed model weights served in place.

— Names exact model and checkpoint but no further detail snapshot-20261022

WAS THIS USEFUL? 👍 👎

Modal

Feedback ⓘ announced · Oct 21, 2026 BLOG ↗ RANK

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

> Modal shipped two new capabilities: Dojo, a production LLM post-training library with built-in observability and CLI tooling, and Sticky Sessions, a decorator-based mechanism for pinning session traffic to containers with session-aware autoscaling.

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

01 **Dojo library for LLM post-training** NEW 95

Modal launched the `modal_dojo` Python library for production LLM post-training in under 100 lines of code, built around `TrainConfig`, `HuggingFaceDataset`, and config.launch(). Training data can be loaded from Hugging Face via `HuggingFaceDataset` using `hf_repo`, `hf_split`, `input_column`, `output_column`, and `input_format` parameters, and custom reward functions can be attached to recipe objects via the `custom_rm_function` parameter. It ships with a built-in observability dashboard covering loss and reward curves, framework-emitted metrics, per-rollout agent trajectories, granular substep profiling, and container logs, plus a CLI for retrieving a run's status, metadata, logs, and agent traces by training run ID.

Kick off a fine-tuning run on a Hugging Face dataset with a custom reward function to train a Qwen3 model on proprietary math reasoning data.

```python
from modal_dojo import (
    HuggingFaceDataset,
    Qwen3_8_27B,
    Qwen3_8_27B_Recipe,
    TrainConfig,
)

model = Qwen3_8_27B()

async def my_reward(args, sample, **kwargs) -> float:
    return float(sample.response.strip() == sample.label.strip())

config = TrainConfig(
    model=model,
    dataset=HuggingFaceDataset(
        hf_repo='my-org/my-dataset',
        hf_split='train',
        input_column='prompt',
        output_column='answer',
        input_format='text',
    ),
    recipe=Qwen3_8_27B_Recipe(
        custom_rm_function=my_reward,
    ),
)

run = config.launch()
print(run.training_run_id)
```

— Names full API surface and includes a runnable training example.

launch-20261021-7b7e73e2

02 **Sticky Sessions with session-based autoscaling** NEW

95

Modal added the @modal.sessioned() decorator to create Modal Servers with strong per-session container affinity, ensuring every request in a session lands on the same container. Sessions are started via Server.sessions.start(idle_timeout=<seconds>), which returns a signed session token encoding routing metadata so the proxy can route requests without centrally stored session state; the token is shareable among multiple clients for collaborative apps and is passed via the `Modal-Authorization: Bearer <session.token>` header for authenticated HTTP requests. Sticky-session Modal Servers also support session-based autoscaling, scaling the container fleet by active session count rather than request volume, with queuing when per-container concurrency is exhausted.

Pin a stateful inference server (e.g. a voice agent or streaming ASR model) to a single container per session, with a 10-minute idle timeout before the session is torn down.

```python
@app.server()
@modal.sessioned()
class Server:
    @modal.enter()
    def start(self):
        subprocess.Popen(["python", "-m", "http.server", "8000"])

@app.local_entrypoint()
def main():
    session = Server.sessions.start(idle_timeout=600)
    headers = {"Modal-Authorization": f"Bearer {session.token}"}
    print(requests.get(Server.get_url(), headers=headers).text)
```

— Names decorator, SDK call, header convention, and gives runnable example.

launch-20261020-35d49e65

WAS THIS USEFUL? 👍 👎

Baseten

Feedback ⓘ elephant-1 and mammoth-1 available on Baseten

NOTES ↗ RANK

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

> Baseten expanded its Model APIs catalog with two new models, elephant-1 and mammoth-1, for text classification and scoring tasks.

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

01 **elephant-1 and mammoth-1 added to Model APIs** NEW 50

Baseten adds `elephant-1` and `mammoth-1` to its Model APIs catalog, enabling text classification, request routing, and content scoring against a rubric in a single multi-question request.

— Names two models and use cases but no API call shown

elephant-1 and mammoth-1 available on B…

WAS THIS USEFUL? 👍 👎

Cloudflare Workers AI

Feedback ⓘ

Agents, Workers AI, AI Gateway - createAI replaces workers-ai-provider and ai-gateway-pr

BLOG ↗ RANK

Cloudflare Workers AI runs open-weight models, including the Clef decision models, as serverless inference on Cloudflare's global network.

> Cloudflare Workers AI shipped a unified `createAI` provider for the Vercel AI SDK that merges Workers AI models, AI Gateway catalog models, dynamic routes and vendor models with fallback, BYOK and stream-resume support, alongside a parallel `createAI` integration for Pi AI, while deprecating the older workers-ai-provider and ai-gateway-provider packages.

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

01 **Unified createAI provider for the AI SDK** NEW 98

`createAI` from `agents/models/ai-sdk` unifies Workers AI models, AI Gateway catalog models, dynamic routes, and vendor models behind the Vercel AI SDK, replacing the deprecated Workers AI provider package. It supports `ai.embedding`, `ai.image`, `ai.speech`, `ai.transcription`, and `ai.reranking` model types, plus `streamText` for streaming output. Requests route through AI Gateway by default, with `id: '<gateway-id>'` on `createAI` to target a named gateway and access catalog models like `anthropic/claude-opus-5.5`. Per-model `fallback` arrays (e.g. `['@cf/zai-org/glm-4.7-flash']`) provide ordered fallback when a primary model is unavailable; `byokAlias`, `zdr`, and `vendorKey` gateway options configure Bring Your Own Key and zero data retention; `dynamic/<route>` model IDs invoke dynamic routes; vendor model objects (e.g. ai(anthropic('claude-opus-5-5'))) route unchanged to AI Gateway's provider-native endpoint; and the binding supports stream resume for catalog and dynamic route streams that drop mid-response. Requires `ai@^7` and `@ai-sdk/provider@^4` alongside the Agents SDK, and an `'ai'` binding in Wrangler configuration.

Wire up a Workers AI model to the AI SDK and generate text in a Worker request handler.

```javascript
import { createAI } from 'agents/models/ai-sdk';
import { generateText } from 'ai';

export default {
  async fetch(request, env) {
    const ai = createAI(env.ai);
    const { text } = await generateText({
      model: ai('@cf/zai-org/glm-4.7-flash'),
      prompt: 'Write a 50-word essay about hello world.',
    });
    return new Response(text);
  },
};
```

Declare the required `ai` binding in Wrangler configuration so `createAI` can reach Workers AI.

```json
{
  "ai": {
    "binding": "ai"
  }
}
```

Generate embeddings, an image, speech, a transcription, and reranked results from the same `createAI` provider instance.

```javascript
import { createAI } from 'agents/models/ai-sdk';
import { embedMany, generateImage, generateSpeech, rerank,
transcribe } from 'ai';

const ai = createAI(env.ai);

// Embeddings
const { embeddings } = await embedMany({
  model: ai.embedding('@cf/baai/bge-base-en-v1.5'),
  values: ['hello', 'world'],
});

// Image
const { image } = await generateImage({
  model: ai.image('@cf/black-forest-labs/flux-1-schnell'),
  prompt: 'A cyberpunk lizard',
});

// Reranking
const { rerankedDocuments } = await rerank({
  model: ai.reranking('@cf/baai/bge-reranker-base'),
  documents: ['a cyberpunk lizard', 'a rainy Tuesday'],
  query: 'Which one is cooler?',
});
```

Route a streaming prompt through AI Gateway with an automatic fallback model if the primary is unavailable.

```javascript
import { createAI } from 'agents/models/ai-sdk';
import { streamText } from 'ai';

export default {
  async fetch(request, env) {
    const ai = createAI({ binding: env.AI, id: 'my-gateway' });
    const result = streamText({
      model: ai('anthropic/claude-opus-5.5', {
        fallback: ['@cf/zai-org/glm-4.7-flash'],
      }),
      prompt: 'Summarise this incident report.',
    });
    return result.toTextStreamResponse();
  },
};
```

— Names every option, endpoint-equivalent flag and binding with runnable examples.

Agents, Workers AI, AI Gateway - create…

02 **Pi AI integration via createAI** NEW 88

`createAI` from `agents/models/pi-ai` (requires `@earendil-works/pi-ai@~1.3.0`) connects Workers AI models to Pi AI's `complete`, `stream`, `streamSimple`, and `completeSimple` entry points. ai.stream() gives event-based streaming, while ai.streamSimple() and ai.completeSimple() accept Pi AI's simple options including `reasoning` level and `maxTokens`. ai.images() and ai.generateImages() generate images with Workers AI models such as `@cf/black-forest-labs/flux-1-schnell`, and optional `contextWindow` / `maxTokens` arguments on the ai() constructor override settings for Workers AI model ids not in Pi AI's catalog. Requests route through AI Gateway, selectable via the `id` option on `createAI` (defaults to `'default'`), with `accountId` and `apiToken` enabling calls from outside a Worker; a new `ai` binding configuration key exposes the binding as `env.AI` in Wrangler.

Generate text in a Worker using a `@cf/` model via Pi AI's `complete` entry point.

javascript

```
import { createAI } from "agents/models/pi-ai";

export default {
  async fetch(request, env) {
    const ai = createAI({ binding: env.AI });
    const reply = await ai.complete(
      ai("@cf/zai-org/glm-4.7-flash"),
      { messages: [{ role: "user", content: "What is
Cloudflare?", timestamp: Date.now() }] }
    );
    const text = reply.content
      .filter((part) => part.type === "text")
      .map((part) => part.text)
      .join("");
    return Response.json({ text });
  },
};
```

Route Workers AI requests through a named AI Gateway instead of the default one.

javascript

```
const ai = createAI({ binding: env.AI, id: "my-gateway" });
```

Add the Workers AI binding to your Wrangler configuration so `env.AI` is available in your Worker.

```toml
[ai]
binding = "AI"
```

— Names entry points, options and binding with runnable code.

Agents, Workers AI, AI Gateway - create…

---

03 **Deprecation of workers-ai-provider and ai-gateway-provider**

DEPRECATED

60

The original `workers-ai-provider` and `ai-gateway-provider` packages for the AI SDK are now feature-frozen and deprecated in favor of the unified `createAI` from `agents/models/ai-sdk`. `createAISearch` from `workers-ai-provider` is replaced by `createAISearchNamespace` from `ai-search-provider`.

— Names the deprecated and replacement packages but gives no migration code.

Agents, Workers AI, AI Gateway - create…

└──▷ **BREAKING ON UPGRADE**

! The existing Workers AI provider package for the AI SDK is deprecated; migrate to `createAI` from `agents/models/ai-sdk`.

WAS THIS USEFUL? 👍 👎

Fireworks AI

Feedback ⓘ

Training SDK 1.2.30: clearer TITO errors and trajectory discard · NOTES ↗ · RANK

Fireworks AI provides hosted inference, model fine-tuning, and deployment APIs for open-weight models.

> Fireworks AI's Training SDK 1.2.30 overhauls TITO error reporting with granular, breaking error codes and new headers, while also adding configurable failure trace capture, multimodal image inputs for sampling, and new observability for discarded trajectories.

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

01 Granular TITO error codes and headers

BREAKING

90

`tito_model_malformed_output` and `tito_upstream_error` are removed and replaced by four new error codes: `tito_sdk_parse_error` (HTTP 422, with stable `parser_reason_code`, finish reason, and token counts), `tito_server_http_error`, `tito_server_stream_error`, and `tito_transport_error` (502/504), so TITO failures identify their actual source. Error responses now also carry `x-fireworks-error-origin` and `x-fireworks-error-code` headers, and requests to a closing or closed trajectory return a new `tito_trajectory_closed` (HTTP 410) response including the reason it ended. Any error-handling code matching on the old error codes will stop matching.

— Names every removed and added error code, header, and status.

Training SDK 1.2.30: clearer TITO error…

02 Image inputs for FiretitanSamplingClient.sample

NEW

88

`FiretitanSamplingClient.sample` and `sample_async` gain an `images=` keyword-only argument accepting a list of base64 data URLs for token prompts, with one placeholder per image, enabling multimodal prompts during agentic RL sampling.

Send an image alongside a text prompt during agentic RL sampling, now that `FiretitanSamplingClient.sample` accepts multimodal inputs.

python

```
result = await client.sample_async(
    prompt=token_prompt,
    images=['<base64_encoded_image>']
)
```

— Exact method and argument named, with runnable example.

Training SDK 1.2.30: clearer TITO error…

03 **Configurable TITO failure trace capture** NEW 85

`TITOSidecar` now accepts a failure_trace_config=TITOFailureTraceConfig(...) parameter to relocate or disable local JSONL failure trace capture; traces default to the `fireworks-tito-traces` directory under the system temp directory and are capped at 256 MiB.

Redirect TITO failure traces to a known path (e.g. a shared debug volume) instead of the system temp directory, so the whole team can inspect failed trajectories.

```python
sidecar = TITOSidecar(
    ...,

failure_trace_config=TITOFailureTraceConfig(directory='/mnt/debug
/tito-traces')
)
```

— Names exact config class, default path and size cap.

Training SDK 1.2.30: clearer TITO error…

04 **Trajectory discard metric and resilient RL loop** IMPROVED 67

Adds a `producer/discard_errors_total` metric reporting TITO failure counts broken down by reason, to surface when most trajectories are being discarded. Unparseable trajectories are now silently discarded by the cookbook RL loop without retrying or halting the run, and parse failures no longer count toward the infrastructure circuit breaker.

— Names metric and behavior change but no config example.

Training SDK 1.2.30: clearer TITO error…

└──▷ **BREAKING ON UPGRADE**

! `tito_model_malformed_output` and `tito_upstream_error` are removed and replaced by `tito_sdk_parse_error`, `tito_server_http_error`, `tito_server_stream_error`, and `tito_transport_error`; any error-handling code that matches on the old names will stop matching.

WAS THIS USEFUL? 👍 👎

Together AI

Feedback ⓘ announced · Oct 19, 2026 BLOG ↗ RANK

Run, train, and serve open-source AI models on Together AI.

> Together AI launched Together Post-Training, a new managed service for fine-tuning and reinforcement learning on dedicated GPU resources, with Tinker API compatibility, multi-LoRA support for both training and serving, custom loss gradients, and new base models.

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

01 **Tinker API drop-in replacement module** NEW 80

`together.post_training.tinker` acts as a drop-in replacement for the Tinker API: replace `import tinker` with `import together.post_training.tinker as tinker` to migrate existing post-training scripts without further changes.

Migrate an existing Tinker API post-training script to Together Post-Training with a single-line change.

python

```
import together.post_training.tinker as tinker
```

— Exact import statement given as runnable migration step. launch-20261019-4a3d0c0c

02 **Built-in and custom training losses** NEW 75

Supports built-in losses including `cross-entropy`, `GRPO`, `CISPO`, and `DPPO`, plus custom training objectives where you compute a custom loss locally and send its gradients to the Trainer for weight updates — giving full control over the training loop including data, environments, agentic harnesses, logging, and external benchmarking steps.

— Names all loss types and the custom gradient mechanism. launch-20261019-4a3d0c0c

03 **Post-training cookbooks repository** NEW 65

Provides post-training cookbooks at `github.com/togethercomputer/post-training-cookbook` covering GRPO on GSM8K, DPO as a custom loss, and multi-LoRA training within a single model resource.

— Named repo path gives a clear starting point for readers. launch-20261019-4a3d0c0c

04 **Multi-LoRA adapter serving in Dedicated Model Inference** NEW 60

Adds multi-LoRA adapter serving in Dedicated Model Inference, allowing adapter swaps within a single GPU deployment instead of requiring separate deployments per run.

— Names the product area and the before/after deployment behaviour.

launch-20261019-4a3d0c0c

THINNER COVERAGE BELOW

05 **Together Post-Training service launch** NEW 45

Together Post-Training is a new managed service for supervised fine-tuning (SFT) and reinforcement learning (RL) using dedicated GPU allocations called Model Resources.

— Names the service and GPU allocation concept but little mechanism.

launch-20261019-4a3d0c0c

06 **LoRA and full-weight fine-tuning** NEW 45

Supports LoRA training with multiple LoRA adapters sharing the same GPU capacity, as well as full-weight fine-tuning for select models.

— States capability without specifics on supported models or limits.

launch-20261019-4a3d0c0c

07 **Model Resource GPU-minute billing** NEW 45

Bills Model Resources per GPU-minute of B200 usage, with an introductory rate of $9 per GPU-hour for the first eight weeks.

— Concrete pricing but no action for the reader to take. launch-20261019-4a3d0c0c

08 **Asynchronous RL across Trainer and Generator** NEW 40

Enables asynchronous RL for high GPU utilization across Trainer and Generator instances, particularly for long agentic rollouts.

— Describes the mechanism but no config detail to act on. launch-20261019-4a3d0c0c

09 **New base models for post-training** NEW 35

Launches with support for Muse Spark Open, GLM 5.3, Kimi K3, and Qwen 3.8-27B as base models for post-training experiments.

— Just a list of model names, no usage detail. launch-20261019-4a3d0c0c

WAS THIS USEFUL?  

AWS Amazon SageMaker AI

Feedback ⓘ October 15, 2026

NOTES ↗ RANK

Amazon SageMaker AI provides managed tools and infrastructure for building, training, fine-tuning, and deploying machine-learning models.

> Amazon SageMaker AI updated the HyperPod cluster instance role policy to support sending OpenTelemetry metrics to CloudWatch.

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

01 OTLP metrics from HyperPod clusters to CloudWatch IMPROVED

55

The `AmazonSageMakerClusterInstanceRolePolicy` managed policy was updated to allow HyperPod cluster instances to emit OpenTelemetry Protocol (OTLP) metrics to Amazon CloudWatch.

— Names exact policy and protocol but no setup steps October 15, 2026

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

OpenWhispr

Feedback ⓘ v1.10.4 NOTES ↗ RANK

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

> OpenWhispr 1.10.4 adds per-app dictation tone control, calendar-linked notes, an in-app docs assistant, and per-scope reasoning effort tuning, alongside smaller model and UI additions.

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

01 **Tone & Style per-app dictation** NEW 70

A new Tone & Style feature lets users select a writing tone per app category (personal messages, work chats, email, etc.) so cleanup rewrites dictation and assistant drafts in that tone. It is off by default and enabled in Settings → Tone & Style.

Keep dictated work messages consistently professional without editing tone manually after each transcription.

📍 In the app, go to Settings → Tone & Style, enable the feature, then assign a tone (e.g. Professional) to the Work chats app category.

— Names exact settings path and default state. v1.10.4

THINNER COVERAGE BELOW

02 **Calendar event linking on notes** NEW 55

Notes can now be linked to a calendar event, pulling in the guest list and join link for that meeting directly into the note.

Tie a note to a meeting so guest names and the join link are available inline — useful when reviewing meeting notes later.

📍 Open a note, select Link to calendar event, then choose the corresponding meeting from the calendar picker to attach guests and the join link.

— Clear steps given but no API/config name. v1.10.4

03 **Per-scope reasoning effort control** NEW 55

Adds a Reasoning effort setting that lets users choose how hard reasoning models think on a per-scope basis, selectable as low, medium, or high in the model settings for the given scope.

Tune response quality vs. speed for a reasoning model by adjusting how much compute it spends per scope.

> 📍 In the model settings for the desired scope, locate `Reasoning effort` and select the appropriate level (low, medium, or high) for that scope.

— Names the setting and its levels with location. v1.10.4

---

04 **wl-clipboard required for Wayland auto-paste** BREAKING

50

On niri, sway, and Hyprland, auto-paste into Wayland apps now requires `wl-clipboard` to be installed.

— Names exact dependency and affected compositors. v1.10.4

---

05 **New model support: Claude 5.5 and Qwen3.8 27B on Groq** NEW

35

Adds support for Claude 5.5 models and Qwen3.8 27B on Groq.

— Names models but no selection steps. v1.10.4

---

06 **In-app docs assistant "Ask about OpenWhispr"** NEW 30

Adds `Ask about OpenWhispr`, a chat interface that answers questions about the app using the docs and the user's current settings.

— No UI path or invocation method given. v1.10.4

---

07 **AI Summary output depth improvement** IMPROVED 30

`Generate AI Summary` now produces the same depth of output as the Detailed Notes template, an improvement over its prior shallower output.

— States before/after but no usage detail. v1.10.4

---

08 **"This is me" speaker identification** NEW 25

Adds a `This is me` option in the speaker picker so users can identify themselves in transcripts.

— Bare feature name, no further mechanism. v1.10.4

---

09 **Sidebar and onboarding UI refinements** IMPROVED 25

Adds searchable overflow for long sidebar folders and onboarding key checks as you type.

— Two thin UI tweaks with no further detail. v1.10.4

└──▷ BREAKING ON UPGRADE

! On niri, sway, and Hyprland, auto-paste into Wayland apps now requires `wl-clipboard` to be installed.

WAS THIS USEFUL? 👍 👎

Ollama

Feedback ⓘ v0.40.8 NOTES ↗ CODE ↗ RANK

Ollama downloads, serves, and chats with local language models through a CLI and REST API.

> Ollama v0.40.8 adds support for running Flint, Recursive's 40B-parameter mixture-of-experts coding model with a large context window, directly via the CLI.

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

01 **Flint coding model support** NEW 86

Adds `ollama run flint` to run Flint, Recursive's 40B-parameter mixture-of-experts coding model, featuring a 262K-token context window, multi-turn tool use, and reasoning support. Currently available on macOS, intended for agentic coding tasks like understanding unfamiliar code, tracing bugs, or implementing features across multiple files.

Run Flint locally on macOS for agentic coding tasks across a repository — understanding unfamiliar code, tracing bugs, or implementing features across multiple files.

```
$ ollama run flint
```

— Names exact command, model size, context length, and platform limit. v0.40.8

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

Mozilla AI Otari

Feedback ⓘ v0.31.0 NOTES ↗ RANK

Open-source, OpenAI-compatible LLM gateway you run yourself. One endpoint for 40+ providers, with virtual keys, budgets, and usage tracking.

> Across 26 releases, Otari shipped a full hosted-providers system with health checks and a migration path, an organization-owned guardrail authoring and enforcement platform, end-to-end agent session tracing, a pricing engine now fed by models.dev, and a sandboxed Files API/code-execution surface — alongside breaking changes to provider configuration, auth, and error envelopes.

└──▷ **WHAT SHIPPED · 28 FEATURES** most completely described first ? what's the number?

01 **Files API and sandboxed code execution** NEW 93

Adds a `code_execution_executor` setting with a code-execution vocabulary for running code via providers or a sandbox, sandbox support for resuming a code-execution container across requests by container ID, and routes provider-native code execution through a port on a hosted provider, falling back to the sandbox when the model has no native executor. Adds `GET /api/v1/files` (paginated) serving both SDKs' Files APIs, including Anthropic's GA Files API shape that refuses the files beta header; adds a `files_sweep_interval_sec` config key to control the file-reclaim sweep (set to `0` to disable); adds an fsspec store and shared file helpers as the backing layer, and SeaweedFS as a self-hosted object store option in Compose; and copies files produced by a provider's code-execution into Otari's own store. Passes attached request files through to the provider's own code execution environment. File storage now allocates a storage reference before writing bytes, and the `code_execution_executor` setting defaults to `auto` on upgrade.

— Names endpoint, config keys and defaults with before/after. v0.7.0 v0.14.0

02 **Organization-owned guardrail authoring and enforcement** NEW 88

Adds an API surface to store, write, look up, serve, publish, rebuild, and report runtime status of an organization's own guardrail definitions, including built-in guardrails by name. Lets organizations mandate a guardrail they defined themselves while blocking deletion of any definition a mandate references, and denies `any_llm` access to an organization's own guardrail store. Primes the guardrail runner at gateway boot and keeps it current as definitions change, running mandated definitions on the request path, and sizes the guardrail thread pool from deployment configuration. Adds an organization guardrails page to the dashboard, with dialogs to configure self-hosted or remote guardrails, checkbox-driven JSON argument entry, per-row test capability, and a

view showing each workspace the guardrails that apply to it. Adds an endpoint to test a mandate against its own guardrails service, plus a matching test action in the dashboard.

— Full mechanism and dashboard path named; no exact endpoint URL. `v0.9.0` `v0.10.0`

---

03 **Guardrail CLI, composition, and policy-check gate controls**

IMPROVED

85

Adds `otari guardrails validate` to check a guardrail offline without a live connection, and supports composing a guardrail from files in the `.otari/guardrails/` directory or per-user from `~/.otari` via the hook layer. Lets a judge cap be set via the CLI and reports any judge that gets skipped, lets a gate's message be optional, and lets a gate accept a guardrail warning it intentionally expects. Runs agent guardrails checks against every pull request via the CLI, adds `no-stranded-docblocks` as a built-in `check_passed` verifier for policy checks and lets a repository author its own `check_passed` verifier. Enforces a policy-check gate that refuses a file read before the file enters the transcript, exposes when each policy-check gate runs and what context it can inspect, and delivers a judge gate's finding directly to the agent rather than only the human operator. Supports Codex as a hook harness alongside Claude Code, evaluates otari hook policies locally by default, runs `otari gates generate` and hook gates concurrently, and ships the agent-side CLI as the `otari-agent` package. Renames Agent Gates to Agent Guardrails throughout the policy-checks subsystem.

— Names every CLI command and directory but spans many releases. `v0.7.0` `v0.10.0` `v0.12.0` `v0.12.1` `v0.14.0` `v0.16.0` `v0.17.0`

---

04 **Web search and fetch tool integrations (any-search/any-fetch)**

NEW

85

Adds the `any-search` and `any-fetch` workspace packages for search and fetch capabilities, serving provider catalogs from both tools and adding fetch instances and connection tests to search-tools. Adds the Mojeek adapter and Exa search/contents adapters (with Exa answer-field checks and live tests in CI) to `any-search`, and the Tabstack extract adapter to `any-fetch`; routes web fetch through `any-fetch` on fetch instances, and leaves web search with `DeepSeek` on any-llm 1.37.1. Adds a web search resolver tool under the `tools` subsystem, in-loop web search support running inside the tool loop on search instances, a direct search endpoint on the any-search path, and configurable default tool policy for workspaces that have no tool policy set. Enables a single request to combine web tools, code execution, and MCP together, reports the gateway's web tool calls on Chat Completions responses, and the playground chat now shows tool calls and supports multiple tools. A workspace's web search limits are now applied as a ceiling in hybrid mode, so requests that previously exceeded workspace limits are now capped.

— Names every adapter and package; mechanism for limits noted. `v0.14.0` `v0.25.0` `v0.26.0` `v0.29.0` `v0.19.0` `v0.20.0` `v0.21.0` `v0.18.0` `v0.30.0`

05 **Explicit provider declarations required in config** BREAKING

85

Deprecated calling an undeclared provider through its environment variable, signaling future removal of that implicit lookup path, then enforced it: Otari now serves only explicitly declared providers, keyed by their config-file name. Provider keys are no longer read from environment variables automatically — each provider must be declared in config, e.g. `openai: {api_key: ${OPENAI_API_KEY}}` for keyed providers, or a bare `ollama:` for keyless providers (Ollama, vLLM, Bedrock, ...). Requests to any undeclared provider return `424 provider_not_configured`.

— Gives exact config example and error code with before/after. v0.28.0 v0.13.0

06 **Hosted providers with catalog and health integration** NEW

83

Adds hosted providers support: configure via `config.yml` and environment variables, served at dispatch and surfaced on the deployment rail dashboard page, backed by a new hosted-providers operator API and service layer with database tables and repositories, and integrated into discovery and health catalog checks. This was extended to named and keyless hosted providers, replacing the Deployment providers page entirely, with a migration path from `config.yml` `providers:` keys.

— Names config keys, API layer and migration path, no exact endpoint. v0.27.0 v0.30.0

07 **Pricing engine and tool-call billing** IMPROVED

81

Adds a distinct pricing kind for search and fetch tool prices and gates in-loop web tools on the price that settles them; prices tool calls by label with reported cost first; and charges completion models priced per request a flat amount per call. Switches the pricing source to a `models.dev` price index, replacing `genai-prices`, and runs the pricing snapshot lifecycle on `models.dev`. Charges deployment-paid inference to the organization's funds, tracks in usage rows whether the deployment paid for a request, and notifies overlays about each usage row a request settles. Direct searches now require pricing to be configured. Surfaces unpriced usage to operators in the dashboard so billing gaps are visible, and lets organizations supply their own web search key for search tool calls. Publishes organization usage on standalone deployments, and returns request ID and inline cost on standalone responses.

— Covers pricing source, flags and billing mechanics across many releases. v0.9.0 v0.13.0 v0.16.0 v0.19.0 v0.26.0 v0.27.0 v0.29.0 v0.30.0 v0.31.0

08 **Gateway HTTP headers for model, retry, and tool control** NEW

81

Adds Otari-Model response header naming the model that served each request, Retry-After response header when a budget policy refuses a request, and Otari-Web-Fetch header to intercept and take over Anthropic's web fetch tool, routing it through Otari's

own fetch handling. Adds the `Otari-Web-Search` HTTP header to the gateway, enabling operators to run a provider search keyword directly via the API. Otari's `X-` prefixed response headers have been renamed; any client code or middleware inspecting those headers must be updated.

— Names every header verbatim with exact behaviour. v0.25.0 v0.19.0 v0.12.0

09 **Idempotency support for completions** NEW 80

Supports an `Idempotency-Key` header on non-streaming completions, returning cached responses on retries and a 409 immediately when the original request is still running. Encrypts stored idempotent responses using `OTARI_SECRET_KEY` to protect replay data at rest.

— Names header, status code and encryption env var precisely. v0.14.0

10 **End-user and budget management for service keys** NEW 78

Adds a cap on how many end users a service key may have, and a service-key-scoped API to delete end users individually by id or in bulk by age. Lets a service key manage its own end users via the budget system without requiring a separate admin credential, and supports end-user IDs that contain '/' in budget routes. Allows a single service key to enroll end users across multiple budgets, bills end users for search usage, and supports lookup by caller-supplied user ID via service keys. Adds TPM (tokens-per-minute) rate limiting counted on tokens used, consistent with LiteLLM behavior, per-user budget limits per minute via the budgets subsystem, and support for serving provider endpoints owned by a workspace or individual user.

— Enumerates every budget/end-user surface but omits exact API paths. v0.16.0 v0.25.0 v0.30.0 v0.31.0

11 **Agent session tracing and Observe UI** NEW 78

Records a trace for every completion served, grouping a session's requests into one trace, and preserves spans that instrumented agents export over OTLP — including booking Claude Platform OTLP spans once at their cache convention. Stores agent traces in a dedicated `traces` domain, exposes a trace API to read agent sessions back programmatically, and adds an Activity page in the web UI with a side panel showing sessions and requests in one list. Adds a Sessions page under Observe for viewing agent traces, with column-header sorting on the Sessions list and search across sessions by any part of their id, name, agent, or user. Groups a Codex thread's requests into a single session, names a session by the client-chosen id, and shows cached input tokens on agent traces.

— Names the traces domain and UI pages but not a specific API path. v0.23.0 v0.24.0 v0.26.0 v0.27.0 v0.30.0 v0.30.1 v0.31.0

12 **Model catalog and offerings management** NEW 73

Enables organizations to offer, price, and switch models on their own provider API keys, adds server-side search across the model catalog and model roster, and shows providers and model makers with their own identifying marks on the dashboard. Allows guests to browse hosted models in the catalog and create an account directly from a model listing. Serves `GET /models` on a hybrid gateway from the control plane, serves providers without a Responses API through chat completions, returns the model name in the rerank response, names each offering's model ID on the model page, and shows model capabilities as a labeled row with explanations in the UI. Refuses managed models clearly where managed credentials are not served.

— Names the `GET /models` endpoint; rest described in prose. v0.7.0 v0.11.0 v0.18.0 v0.26.0 v0.29.0

13 **Routing transparency and subagent model selection** NEW

70

Names the serving provider and fallback in response headers for every routed request, shows automatic fallbacks available to callers in the routing UI, and describes a catalog ID's fallback plan in the Activity view. Enables routing to choose subagent candidates by harness, builds and serves the routing question that selects a subagent's model for dynamic per-subagent routing, and adds `otari-router` as a Claude Code plugin to serve subagent model recommendations directly from the plugin surface. Simplifies creating and reading routing policies in the dashboard, and supports configuring provider SDK retries and skipping them before a failover.

— Headers and UI named but no exact header names given. v0.17.0 v0.18.0 v0.20.0 v0.21.0 v0.26.0 v0.27.0

14 **Authentication and signup security fixes** BREAKING

70

Authenticates a backend with a workload identity token. Auth signup verification now uses only its own password and holds every claim to one rule, breaking existing multi-password or cross-claim signup flows. OAuth sign-in is now keyed on the provider subject rather than the previous identifier, so existing OAuth-linked accounts may fail to match on sign-in after upgrade. The invite response's `accept_link` is now `null` when `mail_sent` is `true`, preventing invitation links from claiming accounts held by other organizations (GHSA-9hhv-84wv-7cx2); code reading `accept_link` unconditionally must handle the null case.

— Names the GHSA advisory and exact field, mechanism partly external. v0.19.0 v0.25.0 v0.29.0

15 **Request tagging for usage and spend attribution** NEW

70

Reads request tags from an Otari-Tags HTTP header for usage tracking, enabling cost/attribution reporting per tagged request. Also records request tags from metadata

and enables querying spend broken down by those tags in the `usage` subsystem.

– Names the header and usage subsystem but no query syntax. v0.17.0 v0.28.0

16 **Per-end-user identifiers forwarded to providers** NEW

69

Adds provider user identifier settings to config, controlling how end-user identifiers are derived, and replaces the caller's `metadata.user_id` with the derived identifier before forwarding requests to providers. Sends the end-user identifier to Meta as `safety_identifier` and to Gemini as a label for inference requests.

– Names exact provider fields but not the config key itself. v0.22.0 v0.27.0

17 **In-product feedback and multi-deployment dashboard** NEW

67

Adds a feedback button in the dashboard top bar for sending feedback to the Otari team, and introduces dashboard seams enabling a single dashboard to span multiple deployments. Adds a `feedback_enabled: false` gateway config key to control in-product feedback collection; in-product feedback is now turned on by default, so deployments that relied on it being disabled must now explicitly disable it.

– Names the exact config key and default-on behaviour change. v0.11.0 v0.12.0 v0.20.0

18 **Budgets and API keys declared in `config.yml`** NEW

67

Supports declaring budgets and API keys in `config.yml`, enabling centralized credential and spend control, and writes a declared budget or key back to `config.yml` only when that file changed it, avoiding spurious overwrites.

– Names the exact config file and write-back behaviour. v0.23.0 v0.30.0

19 **Vendor/model addressing syntax** NEW

65

Supports `vendor/model` syntax to reach a model in the catalog, and `provider:vendor/model` syntax to pin a specific provider.

– Exact syntax given but no further mechanism detail. v0.8.0

20 **Dashboard administration for accounts, keys, and organizations**

IMPROVED

65

Supports inviting several organization members at once from the Organizations UI, and defaults a new API key's expiry time to midnight when created via the dashboard. Allows owners and admins to delete an organization with no usage, displays all account emails in the dashboard sign-in frame, and enables setting a key's own budget directly from the

keys page. Adds pagination and name/email search to the deployment Accounts admin page, and shows each MCP server's id with a copy button in the dashboard. Lets a deployment turn passkeys off and puts OAuth first on sign-in and sign-up.

— Names specific dashboard pages and defaults, light on mechanism. v0.13.0 v0.18.0 v0.26.0 v0.27.0 v0.10.0

21 **API error handling and misrouting diagnostics** IMPROVED

61

Assigns the `invalid_request` error code to request validation failures, surfaces the provider's own failure reason on a completion's 5xx response, and returns a 404 with a redirect pointer from `/v1/*` to `/api/v1` for misrouted API calls.

— Names error code and redirect path precisely. v0.20.0 v0.30.0 v0.31.0

THINNER COVERAGE BELOW

22 **Prometheus metrics now an optional extra** BREAKING

58

`prometheus-client` is now an optional extra rather than a hard dependency — installations that rely on Prometheus metrics must now explicitly install the `prometheus-client` extra.

— Names the exact package and required action. v0.14.0

23 **Health checks for hosted models and MCP servers** NEW

57

Adds scheduled health checks that probe every model on a schedule and email a report of what changed, and adds MCP server health validation that checks stored MCP servers on save and displays their health status. MCP peers that return a response omitting the `servers` key are now refused, where previously this was accepted.

— Describes triggers and outputs but no config key or endpoint. v0.30.0 v0.31.0 v0.12.0

24 **Embeddings request and response enhancements** IMPROVED

50

Forwards `input_type` to embedding providers that accept it, and reports cost and pricing source in embeddings usage responses.

— Names the field but limited mechanism detail. v0.20.0

25 **`provider_specific_fields` passthrough for chat requests** NEW

47

Supports `provider_specific_fields` to copy provider extras through in chat requests.

— Names the field but no example of use. v0.16.0

26 **Messages API errors wrapped in Anthropic envelope** BREAKING

45

The Messages API now wraps every error response in Anthropic's envelope format — clients parsing raw error shapes from the Messages endpoint will receive differently structured responses.

— States the change but no migration guidance given. v0.31.0

27 **Responses API opt-in for Messages** NEW

37

Serves Messages requests through the Responses API on opt-in.

— Brief description with no opt-in mechanism named. v0.29.0

28 **Homebrew formula generated from lock file** IMPROVED

30

Renders the Homebrew formula from the lock file.

— One-line packaging change with no further detail. v0.10.0

└──▷ **BREAKING ON UPGRADE**

! The Messages API now wraps every error response in Anthropic's envelope format — clients parsing raw error shapes from the Messages endpoint will receive differently structured responses.

! Direct searches now require pricing to be configured; previously pricing was optional for this path.

! The Deployment providers page and its backing store have been removed; providers previously configured there must be migrated to hosted providers.

! Provider keys are no longer read from environment variables automatically. Each provider must now be declared in config — e.g. `openai: {api_key: ${OPENAI_API_KEY}}` for keyed providers, or a bare `ollama:` for keyless providers (Ollama, vLLM, Bedrock, ...). Requests to any undeclared provider return `424 provider_not_configured`.

! Pricing now sources from `models.dev` instead of `genai-prices`; any configuration or integrations relying on `genai-prices` data will need to be updated.

! Auth signup verification now uses only its own password and holds every claim to one rule — existing multi-password or cross-claim signup flows will break (PR #2288).

! OAuth sign-in is now keyed on the provider subject rather than the previous identifier — existing OAuth-linked accounts may fail to match on sign-in after upgrade (PR #2236).

! In-product feedback is now turned on by default (previously off); deployments that had relied on feedback being disabled by default will need to explicitly disable it.

! The invite response's `accept_link` is now `null` when `mail_sent` is `true`, preventing invitation links from claiming accounts held by other organizations (GHSA-9hhv-84wv-7cx2); any code that reads `accept_link` unconditionally will need to handle the null case.

- ! A workspace's web search limits are now applied as a ceiling in hybrid mode — requests that previously exceeded workspace limits will now be capped.
- ! File storage now allocates a storage reference before writing bytes; integrations that relied on prior write-then-record ordering will break.
- ! `prometheus-client` is now an optional extra rather than a hard dependency — installations that rely on Prometheus metrics must now explicitly install the `prometheus-client` extra.
- ! The provider's code execution now receives the files a request attaches — existing sandboxed code execution behaviour that assumed no file access will change.
- ! MCP peers that return a response omitting the `servers` key are now refused; previously this was accepted.
- ! Policy-check gate execution timing and visible context has changed — existing gates may behave differently after upgrade.
- ! `Agent Gates` is renamed to `Agent Guardrails`; any configuration, scripts, or documentation referencing the old name must be updated.
- ! File reads are now refused by policy-checks before the file enters the transcript; workflows that previously read files at earlier stages will be blocked.
- ! In-product feedback is enabled by default on upgrade; set `feedback_enabled: false` in gateway config to disable it.
- ! Guardrail composition now loads from `.otari/guardrails/`; existing guardrail layouts may need to be migrated to this directory structure.
- ! Otari's `X-` prefixed response headers have been renamed; any client code or middleware inspecting those headers must be updated.
- ! The `code_execution_executor` setting defaults to `auto` on upgrade; existing configs using `code_execution_executor: provider` continue to forward as before.
- ! `GET /api/v1/files` is now paginated on upgrade, and the sweep reclaims files; set `files_sweep_interval_sec: 0` to disable the sweep.

WAS THIS USEFUL?  

◆ Inference Serving & Optimization

vLLM

Feedback ⓘ announced · Oct 21, 2026 BLOG ↗ RANK

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

> vLLM launches vllm-torchtpu, a new out-of-tree backend plugin that brings full TPU support to vLLM, including six-dimension parallelism, custom Pallas kernels, disaggregated serving, speculative decoding and FP8 KV cache quantization.

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

01 **TPU parallelism and async scheduling flags** NEW 90

Adds `--tensor-parallel-size`, `--data-parallel-size`, `--pipeline-parallel-size`, `--enable-expert-parallel`, `--prefill-context-parallel-size`, `--decode-context-parallel-size`, and `--async-scheduling` flags, covering all six parallelism dimensions and asynchronous scheduling for TPU inference on single-host and multi-host TPU slices.

Serve a large MoE model on a TPU v7x host with data-parallel attention and expert parallelism to maximise throughput, caching compiled graphs so subsequent restarts skip recompilation.

```
$ VLLM_XLA_CACHE_PATH=~/.cache/vllm_xla vllm serve Qwen/Qwen3-Coder-30B-A3B-Instruct-FP8 \
  --tensor-parallel-size 1 \
  --data-parallel-size 8 \
  --enable-expert-parallel \
  --kv-cache-dtype fp8 \
  --max-model-len 16384 \
  --async-scheduling
```

Serve a long-context model on TPU with prefill and decode context parallelism to split the KV cache across devices for sessions that exceed per-device HBM.

```
$ VLLM_XLA_CACHE_PATH=~/.cache/vllm_xla vllm serve meta-
llama/Llama-3-70B-Instruct \
  --tensor-parallel-size 4 \
  --prefill-context-parallel-size 4 \
  --decode-context-parallel-size 4 \
  --max-model-len 65536 \
  --async-scheduling
```

— every flag named with runnable example commands launch-20261020-af0bc9d5

02 **vllm-torchtpu backend plugin launches** NEW 80

vLLM introduces `vllm-torchtpu`, an out-of-tree platform plugin installable via `pip install vllm-torchtpu`, that lets vLLM serve models on Google Cloud TPUs using the TorchTPU stack with no changes to existing model code.

— names plugin and exact install command launch-20261020-af0bc9d5

03 **XLA compiled graph caching via env var** NEW 65

Exposes the `VLLM_XLA_CACHE_PATH` environment variable to persist compiled XLA graph artifacts across restarts, avoiding recompilation on startup.

Serve a large MoE model on a TPU v7x host with data-parallel attention and expert parallelism to maximise throughput, caching compiled graphs so subsequent restarts skip recompilation.

```
$ VLLM_XLA_CACHE_PATH=~/.cache/vllm_xla vllm serve Qwen/Qwen3-
Coder-30B-A3B-Instruct-FP8 \
  --tensor-parallel-size 1 \
  --data-parallel-size 8 \
  --enable-expert-parallel \
  --kv-cache-dtype fp8 \
  --max-model-len 16384 \
  --async-scheduling
```

Serve a long-context model on TPU with prefill and decode context parallelism to split the KV cache across devices for sessions that exceed per-device HBM.

```
$ VLLM_XLA_CACHE_PATH=~/.cache/vllm_xla vllm serve meta-
llama/Llama-3-70B-Instruct \
  --tensor-parallel-size 4 \
  --prefill-context-parallel-size 4 \
  --decode-context-parallel-size 4 \
  --max-model-len 65536 \
  --async-scheduling
```

— env var named and used in both examples `launch-20261020-af0bc9d5`

---

04 **Speculative decoding and prefix caching on TPU** NEW

65

Supports speculative decoding (MTP, EAGLE-3, DFlash, DSpark, n-gram, suffix decoding) and automatic prefix caching, including linear-attention state caching for hybrid models like Qwen3.5 and Kimi K3, via existing vLLM flags.

— names methods and models but references unnamed existing flags `launch-20261020-af0bc9d5`

---

05 **Custom Pallas kernels for attention and MoE** NEW 60

Delivers custom Pallas kernels for Ragged Paged Attention v3 (MHA/GQA), dense and sparse Multi-head Latent Attention (MLA), Gated DeltaNet, fused Mixture-of-Experts with expert parallelism, and FP8/INT8/MXFP4/NVFP4 quantized matmuls.

— names many kernels and formats but no invocation given `launch-20261020-af0bc9d5`

---

06 **Disaggregated prefill/decode and KV cache offload** NEW

60

Enables zero-copy KV cache offloading to host DRAM and disaggregated prefill/decode serving across TPU hosts via TPU Sync and vLLM's KV connector interface.

— mechanism named but no flag or example given `launch-20261020-af0bc9d5`

THINNER COVERAGE BELOW

---

07 **FP8 KV cache quantization flag** NEW 50

Adds `--kv-cache-dtype fp8` support for KV cache quantization on TPU.

Serve a large MoE model on a TPU v7x host with data-parallel attention and expert parallelism to maximise throughput, caching compiled graphs so subsequent restarts skip recompilation.

```
$ VLLM_XLA_CACHE_PATH=~/.cache/vllm_xla vllm serve Qwen/Qwen3-Coder-30B-A3B-Instruct-FP8 \
  --tensor-parallel-size 1 \
  --data-parallel-size 8 \
  --enable-expert-parallel \
  --kv-cache-dtype fp8 \
  --max-model-len 16384 \
  --async-scheduling
```

— flag named and shown in example but no mechanism detail launch-20261020-af0bc9d5

08 **TPU hardware support tiers** NEW 50

Validates TPU v7x (Ironwood) in CI on every pull request, while v6e, v5e, and v5p are available as experimental targets.

— states support tiers but no setup steps launch-20261020-af0bc9d5

NEW SUPPORTED PLATFORM Recursive Flint (40B MoE) model support

announced · Oct 21, 2026 BLOG ↗

vLLM gains day-0 Flint support via `vllm/vllm-openai:flint` image and `vllm serve`, with new `--hf-config-path`, `--tool-call-parser glm47`, `--reasoning-parser glm47`, and `--exclude-tools-when-tool-choice-none` flags, plus Flint-specific Blackwell GPU kernels for faster decode.

Adds `vllm/vllm-openai:flint` Docker image (x86_64 and arm64, CUDA 13.0) for serving Recursive's Flint model with no extra setup.

Supports serving Flint's NVFP4 GGUF checkpoint via `vllm serve RecursiveSI/flint-nvfp4-gguf` with `--hf-config-path` and `--tokenizer` pointing to `RecursiveSI/flint`.

Supports serving Flint's BF16 checkpoint via `vllm serve RecursiveSI/flint` with GLM-4.7-style tool-call and reasoning parsers.

Adds `--tool-call-parser glm47` and `--reasoning-parser glm47` to support Flint's GLM-4.7-style tool calls and thinking blocks.

Adds `--exclude-tools-when-tool-choice-none` flag to drop tool definitions from the prompt when `tool_choice` is `none`.

+2 more

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

liliang-cn CortexDB

Feedback ⓘ v2.136.0 NOTES ↗ RANK

AI memory and a knowledge graph in one SQLite file. Pure Go: vectors, RAG, agent memory, RDF/SPARQL, Cypher, 80+ MCP tools.

> CortexDB's three releases this window focus on deepening its RDF/graph layer with W3C-standard SPARQL, SHACL and OWL 2 RL support alongside graph-engine upgrades, while also tightening scoped API-key access to its toolbox and improving RPC request logging and authorization expressiveness.

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

01 **W3C RDF standards support in graph engine** NEW 55

CortexDB adds W3C-conformant `SPARQL 1.1` / `SPARQL 1.2` query support, `SHACL Core` and `SHACL-SPARQL` validation, `OWL 2 RL` reasoning support, and `RDF/XML` serialization support.

— Names four standards but no usage example or API surface v2.120.0

02 **Graph engine: backdated writes, vectorless nodes, anchored Cypher walks**

NEW 50

The graph engine now supports backdated writes that preserve history, vectorless nodes, and id-anchored Cypher walks that traverse from a node's anchor.

— Three distinct capabilities named but no mechanism detail v2.120.0

03 **Confined API key toolbox access over gRPC and REST** NEW 40

Confined API keys can now invoke toolbox calls within their scope over both `gRPC` and `REST`, extending scoped toolbox access across both transports.

— Names transports but no endpoint or config detail v2.136.0

04 **Key and tool name logging in RPC request log** IMPROVED 40

Adds key and tool name logging to each RPC server request in the request log, improving traceability of toolbox calls.

— Names the log and server but no config key or format v2.133.0

05 **Multi-value fields in authorization scopes** IMPROVED 30

Authorization scopes can now list several values per field, expanding access-control expressiveness.

— Vague on which fields or how configured v2.133.0

06 **Node property indexes and execution graph API** NEW

15

CortexDB v2.120.0 also introduces node property indexes and an execution graph API, per the release summary, with no further detail provided.

— Mentioned only in summary with no elaboration v2.120.0

WAS THIS USEFUL? 👍 👎

Microsoft MarkItDown

Feedback ⓘ v0.1.9 NOTES ↗ RANK

Microsoft MarkItDown converts documents, spreadsheets, presentations, and other files into Markdown for language-model workflows.

> MarkItDown v0.1.9 adds language control for audio transcription and dramatically speeds up HTTP streaming of remote files by enlarging the read chunk size.

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

01 **Larger HTTP streaming chunk size** IMPROVED

60

The HTTP response read chunk size was increased from 512 bytes to 64 KB, enabling practical streaming of large remote files and boosting throughput 128×.

— Concrete before/after numbers but no config flag to act on v0.1.9

THINNER COVERAGE BELOW

02 **Language parameter for audio transcription** NEW

45

A new `language` parameter lets callers specify the source language for speech-to-text conversion when transcribing audio.

— Names the parameter but no values or defaults given v0.1.9

WAS THIS USEFUL? 👍 👎

EverMind AI EverOS

Feedback ⓘ v1.5.0 NOTES ↗ RANK

EverOS provides a local-first memory layer that stores and evolves context for AI agents.

> EverOS v1.5.0 adds asynchronous, non-blocking ingestion with task-status tracking, configurable vector dimensions, richer index-health reporting, and a strict startup check that now refuses to boot on embedding model/dimension mismatches.

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

01 Async /add with task-status APIs and queue config NEW

95

`/add` requests now accept `async_mode: true`, returning `202 Accepted` immediately while a durable `memorize_inbox` SQLite queue runs extraction in the background, with concurrency controlled by `max_concurrency` (default 4). New `GET /api/v2/tasks/{task_id}` and `GET /api/v2/tasks` endpoints (filterable by `session_id` / `status`, paged) track async batch outcomes, and the `[memorize.async_queue]` config section adds `enabled`, `max_concurrency`, and `max_replays` (default 3) keys to control the queue.

Fire-and-forget a large ingestion batch from a high-throughput client without blocking on extraction.

```
$ curl -X POST https://everos.example.com/add \
  -H 'Content-Type: application/json' \
  -d '{"session_id": "sess-42", "async_mode": true, "batches":
[{"text": "..."}]}'
```

Poll the outcome of an async batch to know when extraction completed or failed.

```
$ curl 'https://everos.example.com/api/v2/tasks?session_id=sess-
42&status=failed'
```

Configure the async queue with higher concurrency and a custom replay limit for a busy deployment.

```toml
[memorize.async_queue]
enabled = true
max_concurrency = 8
max_replays = 5
```

— Names endpoints, config keys and defaults with runnable examples. v1.5.0

02 **Startup validation blocks embedding model/dimension mismatches**

80

BREAKING

Startup now refuses to start when the configured embedding model's dimension count mismatches `dim` or when the index was built by a different model, naming the values and pointing to `everos cascade rebuild`; the CLI exits with code 3. Any deployment with a mismatched model or dimension will fail to start after upgrading.

— Gives exact exit code and remediation command, clear migration note. v1.5.0

03 **Index-health reporting in /health and cascade status** IMPROVED

75

`GET /health` now includes a `memorize_queue` block (loop alive, backlog, failed, restarts) and a `cascade.index_rows` / `cascade.index_emptied` snapshot to surface queue state and emptied-index conditions. `cascade status` prints a live row count per business table and warns when all tables are empty, distinguishing a deleted index from a healthy one.

— Describes exact fields added but no sample output or call shown. v1.5.0

04 **Configurable embedding vector dimensions** NEW

60

Adds an `[embedding] dim` config key to set the vector column width instead of a hardcoded 1024, enabling embedding models of any output width.

— Names the config key but no usage example given. v1.5.0

└──▷ **BREAKING ON UPGRADE**

! Startup now exits 3 (instead of proceeding) when the configured embedding model returns fewer dimensions than `dim`, or when the index records a different model than the one configured — any deployment with a mismatched model or dimension will fail to start after upgrading.

WAS THIS USEFUL? 👍 👎

Hugging Face Sentence Transformers

Feedback ⓘ

v6.2.0 NOTES ↗ RANK

Sentence Transformers trains and runs embedding, retrieval, and reranking models, and is the reference implementation behind most open embedding models.

> Sentence Transformers v6.2 adds model merging via Arcee AI's mergekit across all four encoder types, automatic distributed evaluation during DDP training, and lower-memory training losses, alongside breaking changes to quantization calibration and checkpoint module-path validation.

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

01 **Model merging across all encoder types** NEW 95

`SentenceTransformer.merge`, `SparseEncoder.merge`, `CrossEncoder.merge`, and `MultiVectorEncoder.merge` combine fine-tuned checkpoints without retraining via Arcee AI's mergekit, supporting methods including `linear`, `slerp`, `ties`, and `dare`. The call accepts `models`, `method`, `parameters`, `base_model`, `device`, `dtype`, and `out_dtype` arguments to control merge inputs, strategy, precision, and output dtype, and works across classification heads, SPLADE models, and token projection layers; install via `pip install 'sentence-transformers[merge]'`, which pulls in `mergekit>=0.1.5`.

Combine two fine-tuned MiniLM checkpoints into a single model with a weighted average, then save and encode with it.

python

```python
from sentence_transformers import SentenceTransformer

inputs = [
    "sentence-transformers/all-MiniLM-L6-v2",
    "sentence-transformers/multi-qa-MiniLM-L6-cos-v1",
]

merged = SentenceTransformer.merge(
    models=inputs,
    method="linear",
    parameters={"weight": [0.6, 0.4]},
)
merged.save_pretrained("merged-MiniLM-L6")

embeddings = merged.encode(["What is model merging?", "Combining compatible model weights."])
print(merged.similarity(embeddings, embeddings))
```

Merge using TIES with a shared base checkpoint, computing in float32 but storing the result in bfloat16 to reduce memory footprint.

```python
from sentence_transformers import SentenceTransformer

inputs = [
    "org/finetuned-model-a",
    "org/finetuned-model-b",
]

merged = SentenceTransformer.merge(
    models=inputs,
    method="ties",
    base_model=inputs[0],
    parameters={"density": 0.5},
    dtype="float32",
    out_dtype="bfloat16",
)
merged.save_pretrained("merged-ties-bf16")
```

Merge two compatible MiniLM fine-tunes with a weighted average to produce a single deployable checkpoint — no retraining required.

```python
from sentence_transformers import SentenceTransformer

models = [
    "sentence-transformers/all-MiniLM-L6-v2",
    "sentence-transformers/multi-qa-MiniLM-L6-cos-v1",
]

merged = SentenceTransformer.merge(
    models,
    method="linear",
    parameters={"weight": [0.6, 0.4]},
)
merged.save_pretrained("merged-MiniLM-L6")
```

— Full method list, args, and install extra named with runnable examples v6.2.0

02 **FP32 fallback layers in CrossEncoder** NEW 75

`CrossEncoder` adds a `keep_in_fp32_modules` argument that computes selected linear layers in FP32 while keeping a reduced-precision backbone, reducing BF16/FP16 score ties.

Run a CrossEncoder reranker in BF16 with the classifier head kept in FP32 to avoid score ties without a full-precision model.

```python
from sentence_transformers import CrossEncoder

model = CrossEncoder(
    "cross-encoder/ms-marco-MiniLM-L6-v2",
    device="cuda",
    model_kwargs={"dtype": "bfloat16"},
    keep_in_fp32_modules=["0.model.classifier"],
)

scores = model.predict([("What is the capital of France?", "Paris is the capital of France.")])
print(scores)
```

— Named argument and purpose with a runnable example v6.2.0

---

03 **Quantization calibration from single vectors removed** BREAKING

75

Inferring `int8` or `uint8` scalar quantization calibration ranges from a single vector now raises `ValueError`. Encode floats first, then call `quantize_embeddings` with representative `calibration_embeddings` or explicit `ranges`.

— Exact error and migration path with function name given v6.2.0

---

04 **Default output normalization in HierarchicalTokenPooling**

65

BREAKING

`HierarchicalTokenPooling` now normalizes output tokens by default, including protected tokens; set `normalize_embeddings=False` on the pooling module to opt out. Models whose saved configurations predate this release retain their previous behavior.

— Exact flag and migration behavior given v6.2.0

---

05 **Automatic distributed evaluation during DDP training** NEW

65

Evaluator encode(), predict(), encode_query(), encode_document(), and rank() calls are now distributed across DDP ranks automatically during training, with rank zero aggregating results and broadcasting metrics.

— Mechanism described but no config example v6.2.0

---

06 **Lower-memory training losses** IMPROVED 65

`BatchSemiHardTripletLoss` and `BatchAllTripletLoss` reduce intermediate memory from cubic to quadratic; `CoSENTLoss` and `AnglELoss` (and their sparse counterparts) reduce intermediate storage to linear via sorted suffix reductions.

— Named losses and complexity change, no usage example v6.2.0

THINNER COVERAGE BELOW

07 **Finer compression control in HierarchicalTokenPooling** IMPROVED 55

`HierarchicalTokenPooling` now accepts fractional `pool_factor` values for finer compression control, and adds optional `fastcluster` linkage for faster clustering.

— Names the args but gives no usage example v6.2.0

08 **Python 3.14 and huggingface_hub 2.x support** NEW 50

Adds support for Python 3.14 and `huggingface_hub` 2.x, while retaining Python 3.10 and Hub 1.3.0 as minimums.

— Exact version ranges named, no usage steps v6.2.0

09 **Relative-only module paths in checkpoints** BREAKING 50

Module paths in checkpoints must now be relative and cannot contain `..` path components, even when `trust_remote_code=True` is set.

— States the restriction but no migration guidance v6.2.0

10 **Faster model loading** IMPROVED 5

Model loading is now faster.

— No mechanism, numbers, or scope given product docs

└──▷ **BREAKING ON UPGRADE**

! Inferring `int8` or `uint8` scalar quantization calibration ranges from a single vector now raises `ValueError`; encode floats first, then call `quantize_embeddings` with representative `calibration_embeddings` or explicit `ranges`.

! `HierarchicalTokenPooling` now normalizes output tokens by default (including protected tokens); set `normalize_embeddings=False` on the pooling module to opt out — models whose saved configurations predate this release retain their previous behavior.

! Module paths in checkpoints must be relative and cannot contain `..` path components, even with `trust_remote_code=True`.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261022 NOTES ↗

RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's latest snapshot adds AI-powered search across runs, traces, and threads, replaces the single organization manage permission with granular role-based permissions, and overhauls the deployment page's settings UI, alongside new Agent Auth GitHub App token support, a `connections.get` `-based` auth flow, and several breaking changes to Teams API access, deployment headers, and tunnel-hosted authorization endpoints.

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

01 Deployment page overhaul: settings, tabs, headers, deletion

80

BREAKING

Deployment settings now open in a side pane with General, Source, Preview Builds, LLM Gateway, and Scaling sections, all saved together in one Save action. The separate Revisions tab on the Deployment page is removed, with bookmarks to tabs after Overview now opening the tab to their right instead; custom headers for deployments have moved out of the Settings dialog into the Connect dialog, so headers previously configured in Settings no longer appear there; and deleting a deployment now requires typing permanently delete to confirm, which immediately removes the database and metadata.

— Full before/after for four UI changes including exact confirmation text

snapshot-20261022

02 Granular organization permissions replace single manage role

75

BREAKING

The single broad organization manage permission is replaced with granular permissions covering `members`, `access control`, `authentication`, `security policies`, `billing`, `usage controls`, `retention`, `data planes`, `workspace creation`, OAuth applications, and `GitHub integrations`.

— Names all 11 new permissions but no exact admin UI path snapshot-20261022

03 Traces section in Deployment Monitoring NEW

60

Deployment Monitoring adds a Traces section with trace count, latency, and error rate charts, aligned to the same time picker as server metrics.

— Describes the new charts but not exact navigation snapshot-20261022

04 Teams API access gated by can_use_teams setting

BREAKING

60

The Teams APIs now return a 403 error unless the organization has the `can_use_teams` setting enabled.

— Names the exact setting but no endpoint list or migration steps snapshot-20261022

THINNER COVERAGE BELOW

05 AI search for runs, traces, and threads NEW

55

Tracing project pages now support AI-powered search across runs, traces, and threads. The feature is enabled by org admins in organization settings, including for personal orgs.

— Names the surface and who enables it, but no mechanism or UI path snapshot-20261022

└──▷ BREAKING ON UPGRADE

- ! The separate Revisions tab on the Deployment page is removed; bookmarks pointing to tabs after Overview now open the tab to their right instead.
- ! Custom headers for deployments are moved out of the Settings dialog into the Connect dialog; headers previously configured in Settings are no longer shown there.
- ! The Teams APIs return 403 unless the organization has the `can_use_teams` setting enabled.
- ! An authorization endpoint placed behind a LangSmith tunnel is rejected at provider creation and update with a 400 error.
- ! Deleting a deployment now requires typing `permanently delete` to confirm, and immediately removes the database and metadata.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.114 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik's biggest addition this window is online evaluation rules — LLM-as-a-Judge, decision-model, and Python code evaluators that auto-score production traces, threads, and spans — alongside a richer rule editor, Diagnostics project guidance with false-positive feedback, Redis Sentinel support, and OpenTelemetry project routing.

└──▷ **WHAT SHIPPED · 12 FEATURES** most completely described first ? what's the number?

01 Project guidance and issue feedback in Diagnostics NEW

95

Adds Project guidance to Diagnostics, letting teams write plain-language instructions (what the agent does, intended behavior, areas to skip) that every later daily or on-demand scan reads before raising findings. Adds a Close issue workflow with two reasons — Resolved (finding stays eligible for future scans) and Not useful (teaches later scans to skip that class of finding, with an optional reason) — surfaced on a Closed issues page with per-reason tabs and a Reopen action. Both features are hidden and inactive until the `TOGGLE_AGENT_INSIGHTS_GUIDANCE_ENABLED=true` environment variable is set and the service restarted. Editing guidance offers Save and run diagnostic (runs immediately, consuming Ollie credits) or Save guidance (waits for the next scheduled run), and the Diagnostics page shows a banner when guidance has changed since the last run.

Enable project guidance and the Not useful false-positive feedback before starting the Diagnostics service, so scans can be steered by team-written instructions.

```env
TOGGLE_AGENT_INSIGHTS_GUIDANCE_ENABLED=true
```

— Names the exact env var, UI pages, and save options. product docs

02 Evaluator rule configuration and editor sidebar IMPROVED

90

Adds a Score row under trace scope to target `production traces`, `experiment traces`, or `all traces` (filters and sampling rate apply only to production traces); adds a Run online evaluation rules dialog to manually run evaluators against selected traces, threads, or spans from the Logs view, bypassing sampling; adds thread scope enrichment where `{{thread}}` expands to every trace and span with type, input, output, and metadata; adds a field-level variable picker triggered by `{{` listing paths

from recent traces/spans with dot and bracket notation; adds a `Clone` action to duplicate an evaluator row with modified settings; adds an `Enabled` toggle in the evaluator footer to create a paused evaluator; and ships a new online evaluation rule editor sidebar in the frontend for configuring these rules.

— Enumerates every named UI control and variable syntax. `2.2.114`

---

03 **LLM-as-a-Judge evaluator with agent mode and multimodal input**

NEW

85

Introduces an LLM-as-a-Judge evaluator with an agent-as-a-judge mode featuring adaptive truncation, bounded tool-call exploration, and a per-evaluation Cost limit (USD) cap, and adds multimodal support — image URLs, base64 images, audio, and video inserted via URL or variable (e.g. `{{input.image_url}}`) — in trace and span scope.

— Names the variable syntax and cost cap mechanism. `product docs`

---

04 **Code evaluator using Python BaseMetric** NEW

80

Introduces a code evaluator that runs a Python `BaseMetric` subclass with fixed `score` method parameters filled from each trace, thread, or span, and adds a `spans=None` parameter to the `score` method so span data is fetched lazily only when declared, keeping evaluations efficient.

— Names the exact subclass and method signature. `product docs`

---

05 **Evaluator editor migrates existing prompts and variable mappings**

75

BREAKING

Existing evaluators with multi-message judge prompts are combined into a single prompt on save in the new editor, with media in a system message removed because it was never sent to the model. Existing evaluators with `{{question}}` `-style` variable mappings are rewritten inline as full paths (e.g. `{{input.messages[0].content}}`) when opened and saved in the new editor.

— Gives exact before/after variable and prompt rewriting behavior. `product docs`

---

06 **OpenTelemetry trace routing by project name** NEW

75

Routes OpenTelemetry traces to a specific project using the `opik.project_name` attribute on incoming spans.

— Names the exact span attribute a reader can set. `2.2.111`

07 **Decision model evaluator via Jev/OpenRouter** NEW 60

Introduces a decision model evaluator backed by Jev (TypeSafe AI's classifier) via OpenRouter, returning per-question Boolean scores and probabilities in a single call.

— Names the backing classifier and provider, no usage steps. product docs

THINNER COVERAGE BELOW

08 **Convert to standard parameters migration for code evaluators** IMPROVED 50

Adds a Convert to standard parameters migration button for code evaluators created with custom argument mappings, rewriting `score` to use standard parameters.

— Names the exact button but no further detail. product docs

09 **SageMaker-signed EventBridge events** IMPROVED 45

Signs EventBridge `PutEvents` calls with SageMaker customer credentials for secure event delivery.

— Names the API call and credential source, no further mechanism. 2.2.111

10 **Internal trace toggle in thread Messages view** NEW 45

Adds a UI toggle in thread Messages view to hide internal traces, with an internal badge marking them.

— Names the exact UI location and badge. 2.2.111

11 **Online evaluation rules for production traces, threads, and spans** NEW 40

Introduces online evaluation rules (evaluators) that automatically score traces, threads, or spans as they are logged, using LLM-as-a-Judge, decision models, or Python code.

— States the concept but no configuration mechanism. product docs

12 **Redis Sentinel support for self-hosted deployments** NEW 25

Adds Redis Sentinel support for high-availability Redis deployments in self-hosted Opik installations.

— Names the feature only, no configuration detail. 2.2.114

└──▷ **BREAKING ON UPGRADE**

! Existing evaluators with multi-message judge prompts are combined into a single prompt on save in the new editor; media in a system message is removed because it was never sent

to the model.

! Existing evaluators with `{{question}}` `-style` variable mappings are rewritten inline as full paths (e.g. `{{input.messages[0].content}}`) when opened and saved in the new editor.

WAS THIS USEFUL? 👍 👎

UK AI Security Institute Inspect AI

Feedback ⓘ

0.3.283 CODE ↗ RANK

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

> Inspect AI 0.3.283 adds sentinel argument passing and richer sentinel call metadata, Meta prompt-cache retention, bridged host tool-call recording, and Scout trust_content scan display, alongside breaking changes to dataset checkpoint fields and Perplexity provider support.

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

01 **Dataset checkpoint field breaking change** BREAKING 95

Dataset loaders now read a top-level `checkpoint` field as the sample's checkpoint configuration and raise when it is not valid. Datasets using `checkpoint` for unrelated data must pass FieldSpec(checkpoint=None) to skip the field, or FieldSpec(checkpoint="<column>") to read checkpoint settings from another column; dataset records can no longer set the checkpoint `trigger`, which must now be set via Task(checkpoint=CheckpointConfig(trigger=...)) or eval(checkpoint=...).

— Names exact fields, classes and migration path 0.3.283

02 **Sentinel argument passing and call metadata** NEW 70

Adds a `--sentinel-arg` flag to pass arguments to a registered sentinel at invocation time. Sentinel tool steps now also carry the id of the message that initiated the call and the request's `tool_choice`.

— Names exact flag and fields but no usage example 0.3.283

THINNER COVERAGE BELOW

03 **Meta provider prompt cache retention** IMPROVED 50

Sends a prompt cache key and requests 24-hour cache retention for Meta (meta) provider calls.

— Explains behaviour and duration but no config example 0.3.283

04 **trust_content scan results in Scout** NEW 45

Adds `trust_content` recording alongside in-eval scan results so Scout displays them as plain text.

— Names the field but no steps to use it 0.3.283

05 **Bridged host tool-call recording** NEW 40

Records bridged host tool calls as tool events in the eval transcript.

— States what changed but lacks mechanism detail 0.3.283

06 **Perplexity provider migrated to Agent API** BREAKING

40

Perplexity provider support has been updated to the Agent API following Sonar's migration; existing Perplexity request configurations may break.

— Flags breakage but gives no migration steps 0.3.283

└──▷ **BREAKING ON UPGRADE**

! Dataset loaders now read a top-level `checkpoint` field as the sample's checkpoint configuration and raise when it is not a valid one. Datasets that use `checkpoint` for unrelated data must pass FieldSpec(checkpoint=None) to skip the field, or FieldSpec(checkpoint="<column>") to read checkpoint settings from another column. Dataset records can no longer set the checkpoint `trigger`; set it with Task(checkpoint=CheckpointConfig(trigger=...)) or eval(checkpoint=...) instead.

! Perplexity provider support has been updated to the Agent API following Sonar's migration; existing Perplexity request configurations may break.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ arize-phoenix-v20.24.0 NOTES ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Phoenix's biggest change this window is breaking: classification evaluators now run on AI SDK decision models via `experimental_decide`, alongside new tracing support for Claude Managed Agents and Atomic Agents, two new playground models, OTLP server log export, and frontend error visibility in the server log.

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

01 **AI SDK decision model support in Evals** BREAKING 95

Classification evaluators, including every built-in, can now run on AI SDK decision models such as openai.decisionModel('gpt-6-luna'), with labels sourced from `experimental_decide` and label probabilities plus provider confidence surfaced in `metadata`; refusals throw `Experimental_DecisionRefusalError`. Adds an `isDecisionModel` export and `DecisionModel` type, and decide spans now record the decision state, question, and answer. The release pins `ai` to `7.0.137` and `@ai-sdk/otel` to `1.0.137` for decision API stability, and requires `@ai-sdk/openai` 4.0.89 or later for plain-text prompt rendering with OpenAI decision models. Breaking: evaluation models such as TypeSafe's Jev now go through `experimental_decide` instead of the previous evaluation model path, with message prompts joined into plain text instead of serialized as JSON state; using `@ai-sdk/openai` earlier than 4.0.89 with OpenAI decision models causes the state to be received as JSON-encoded text instead of plain text.

Run a built-in classification evaluator against an OpenAI decision model to get structured labels and confidence metadata without writing custom parsing logic.

```javascript
import { isDecisionModel } from '@arizeai/phoenix-evals';
import { openai } from '@ai-sdk/openai';

const model = openai.decisionModel('gpt-6-luna');
console.log(isDecisionModel(model)); // true
// Pass model to any built-in evaluator — label comes from
experimental_decide,
// probabilities and provider confidence appear in metadata.
```

— Names exact exports, pinned versions, error types and a runnable code sample.

`@arizeai/phoenix-evals@2.7.0`

02 **Tracing for Claude Managed Agents and Atomic Agents** NEW 62

Phoenix adds tracing for Claude Managed Agents sessions from a Claude Console destination and the Claude SDK, and tracing for Atomic Agents apps via the framework's built-in OpenInference instrumentor.

— Names both frameworks and their instrumentation paths but no setup steps. product docs

THINNER COVERAGE BELOW

03 **Two new playground models with cost tracking** NEW 55

Adds Gemini 4 Argon and Meta Muse Wonder 1.0 to the Phoenix playground, both with built-in cost tracking.

— Names exact models and the playground location to find them. product docs

04 **OTLP server log export** NEW 40

Phoenix now exports server logs over OTLP, correlated with server traces for unified diagnostics.

— Names the export protocol but gives no config or endpoint details. product docs

05 **Frontend errors surfaced in server log** IMPROVED 40

Phoenix v20.24.0 reports uncaught frontend errors to the server log, making browser-side failures visible in backend diagnostics.

— Describes the behavior but no configuration or location specifics.

arize-phoenix-v20.24.0

└──▷ BREAKING ON UPGRADE

! Evaluation models such as TypeSafe's Jev now go through `experimental_decide` instead of the previous evaluation model path; message prompts are joined into plain text instead of being serialized as JSON state.

! Using `@ai-sdk/openai` earlier than 4.0.89 with OpenAI decision models causes the state to be received as JSON-encoded text instead of plain text.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261022 NOTES ↗ RANK

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

> Braintrust's October 2026 release adds Cursor agent tracing, human-review queuing and dataset tooling in its MCP server and Loop automations, a new coding agent insights dashboard, and application logging support in the TypeScript SDK.

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

01 Cursor agent session tracing NEW

85

Run `bt trace enable cursor` (requires `bt` v0.26.0+) to trace Cursor agent sessions — turns, model calls, tool calls, and subagents — from both the Cursor CLI and desktop app.

Enable tracing for all Cursor agent sessions so turns, model calls, and tool calls appear in Braintrust.

```
$ bt trace enable cursor
```

— Runnable command with version requirement and clear scope snapshot-20261022

02 Application logs in TypeScript SDK NEW

75

The TypeScript SDK (v3.39.0+) gains application logs: emit via severity methods info() and error() on a logger, or forward existing `console` output with wrapConsole(), both inspectable alongside traces.

Forward all existing console output to Braintrust application logs in TypeScript without changing every call site.

```typescript
import { wrapConsole } from 'braintrust';
wrapConsole();
```

— Code example and method names given, but no UI path snapshot-20261022

03 MCP human-review queuing for logs and datasets NEW

65

The Braintrust MCP server can send project log traces, experiment traces, or dataset rows to human review and optionally assign them to reviewers; Loop automations can

queue rows for human review on a schedule when granted write tool permissions (requires data plane v2.19.0+).

— Clear capability and version gate but no concrete command `snapshot-20261022`

04 **Coding agent insights dashboard** NEW 65

A new dashboard at `Dashboards > Coding agent insights` shows usage by user, API key, model, tool, and branch with cost breakdowns, plus a daily Loop automation for recurring problem detection (requires data plane v2.13.0+).

— UI path and version given, but no command `snapshot-20261022`

THINNER COVERAGE BELOW

05 **Copy log rows into datasets via MCP** NEW 55

The Braintrust MCP server and Loop can copy existing project log rows into datasets, preserving input, metadata, tags, and source-log links without passing full contents through the assistant context (requires data plane v2.18.0+).

— Describes mechanism but no actionable step or command `snapshot-20261022`

└──▷ **ALSO FROM THESE RELEASES**

Push an observability template and forcibly replace the topic map binding in the selected Topics automation in the destination project.

```
$ bt observability template push --force
```

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI Red Teaming & Testing

AI-Infra-Guard

Feedback ⓘ v4.6.7 NOTES ↗ RANK

AI-Infra-Guard is an AI red teaming platform that scans AI infrastructure for CVE vulnerabilities, MCP/Agent skill risks, jailbreak weaknesses, and API relay security issues.

> AI-Infra-Guard's Agent-Scan module added support for a new protocol stream and improved how prompts and skills are passed to the scanner.

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

01 **AG-UI/CopilotKit protocol support in Agent-Scan** NEW

40

Agent-Scan now supports AG-UI (CopilotKit) protocol streams, expanding the set of agent communication protocols it can scan.

— Names protocol but no mechanism detail v4.6.7

02 **Prompt and skill forwarding to scanner** IMPROVED

35

Agent-Scan now forwards the prompt and selected skills to the Agent-Scan scanner, part of its scan pipeline.

— States behavior change but lacks config/flag detail v4.6.7

WAS THIS USEFUL? 👍 👎

iFixAi

Feedback ⓘ v4.2.1 NOTES ↗ RANK

Independent Auditing of AI Agents. Run by human or the agent itself, to answer the most crucial question in the AI Agent Economy.

> iFixAi's v4.2.0 and v4.1.0 releases add two new auditing subcommands (`trajectory` and `policy-lab`), new AI gateway provider routing, a much faster default inspection suite, and a round of stricter run validation and CI exit-code changes.

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

01 **Faster default essential inspection suite** BREAKING — 95

`ifixai run` now executes a 10-inspection `essential` suite (`B01` , `B08` , `P01` , `B12` , `B09` , `B27` , `B06` , `B16` , `B18` , `B20`) by default instead of all 60, making roughly 6x fewer agent and judge calls while still producing a grade. The `--full-run` flag restores the previous behavior of running all 60 inspections.

— Names exact inspection IDs, call-count reduction, and the opt-out flag. v4.2.0

02 **One-time free-audit prompt with opt-out** NEW — 90

Adds a one-time pre-run prompt on the first interactive `ifixai run` per machine offering a free audit at `https://www.ifixai.ai/` ; never shown in CI, non-TTY, or `--dry-run` environments. The `--no-promo` flag or `IFIXAI_NO_PROMO=1` environment variable suppresses it.

— Names both the suppression flag and environment variable. v4.1.0

03 **ifixai trajectory subcommand** NEW — 80

Adds `ifixai trajectory` to check a recorded sequence of agent actions against a business policy — covering approval reuse, out-of-scope delegation, budget overruns, and wrong citations. It reads JSONL or OTLP and supports offline replay.

— Names the command, input formats, and policy checks it covers. v4.2.0

04 **ifixai policy-lab subcommand** NEW — 75

Adds `ifixai policy-lab` to detect harmful tool-combination sequences whose individual permissions are each allowed, and suggests the minimum permission removal that preserves legitimate goals; runs fully offline.

— Names the command and its exact remediation mechanism. v4.2.0

05 **Stricter CI exit codes for scoped runs** BREAKING — 75

CI exit codes changed: RUN INVALID now always exits 2; a partial run exits 1; a scoped run without `B01`, `B08`, or `P01` exits 0 only if every selected inspection passed, otherwise it exits 2.

— Specifies exact exit codes per run scenario for CI integration. v4.2.0

06 **Stricter pre-run input validation** BREAKING 75

Configuration values previously accepted silently — malformed configs, rubrics, and numeric limits — are now hard errors rejected before the run starts. An invalid `--min-score` value (NaN, Inf, or outside 0–1) is now a usage error that aborts before the run starts, and a resumed run whose endpoint or system prompt has changed is now refused rather than continuing with stale results.

— Names the `--min-score` flag and each validation surface it hardens. v4.2.0 v4.1.0

07 **New AI gateway and cloud providers** NEW 70

Adds `--provider vercel` and `--provider cloudflare` flags to route inspection traffic through Vercel AI Gateway and Cloudflare AI Gateway respectively, and AtlasCloud now appears in guided setup.

— Names both provider flags and the guided-setup addition. v4.2.0

THINNER COVERAGE BELOW

08 **Re-scored results for v4.0.0 runs** BREAKING 50

Scores from v4.0.0 runs may drop or show more unscored probes on re-evaluation because over-credited judge verdicts are now correctly scored.

— Names the version affected and the underlying scoring bug fixed. v4.1.0

09 **Ctrl-C now aborts interactive prompts** BREAKING 20

Ctrl-C at an interactive prompt now aborts instead of accepting the default value.

— Single-sentence behavior change with no further mechanism given. v4.2.0

NEW DETECTION V11 Guilt Tripping inspection

UPDATED COMPLIANCE FRAMEWORK Inspection catalogue expansion

└──▷ BREAKING ON UPGRADE

! `ifixai run` now runs only 10 inspections by default (the `essential` suite); workflows relying on all 60 running automatically must add `--full-run`.

! CI exit codes changed: RUN INVALID now always exits 2; a partial run exits 1; a scoped run without B01, B08, or P01 exits 0 only if every selected inspection passed, otherwise exits 2.

! Configuration values that were previously accepted silently are now hard errors — malformed configs, rubrics, and numeric limits are rejected before the run starts.

! Ctrl-C at an interactive prompt now aborts instead of accepting the default.

! Scores from v4.0.0 runs may drop or show more unscored probes on re-evaluation because over-credited judge verdicts are now correctly scored.

! An invalid `--min-score` value (NaN, Inf, or outside 0–1) is now a usage error that aborts before the run starts.

! A resumed run whose endpoint or system prompt has changed is now refused rather than continuing with stale results.

v4.2.1 NOTES ↗ LATEST STABLE

V11 results surface in JSON as `findings_only: true` with no score or threshold; report-only findings lead all output formats; `ifixai compare` uses verdict.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v4.2.1 https://github.com/ifixai-ai/iFixAi.git
# already have the repo? check out this version:
$ git checkout v4.2.1
```

V11 findings appear in the JSON output under `findings`, with `findings_only: true` and no `score`, `threshold`, `passing`, or confidence-interval fields.

Report-only findings (such as V11) now surface at the top of markdown, HTML, JSON, and terminal output.

`ifixai compare` supports V11 comparison by verdict rather than by percentage.

Every judge's vote is retained in multi-judge runs, including judges that returned no verdict.

WAS THIS USEFUL? 👍 👎

◆ AI/LLM Security

cplt

Feedback ⓘ 2026.10.21-101049-3b2c077 NOTES ↗ RANK

Sandbox for AI coding agents. Runs Copilot CLI, Claude Code, OpenCode, Gemini CLI, Antigravity, Pi, goose or a plain shell inside a kernel-level sandbox, with git and gh guards and sandbox policy committed to the repository.

> cplt adds a new sandbox permission allowing AI agents to use rootless Podman/Docker on Linux without escaping kernel-level sandbox enforcement.

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

01 **Rootless Docker/Podman sandbox permission** NEW 80

A new `allow_rootless_docker` sandbox permission can be set in `.cplt.toml` to grant a sandboxed agent access to rootless Podman/Docker on Linux, without having to disable or escape the Landlock/seccomp-BPF kernel sandbox.

Allow a sandboxed AI agent to invoke rootless Podman or Docker on Linux without disabling kernel enforcement.

toml

```toml
[sandbox]
allow_rootless_docker = true
```

— Names exact config key, file and mechanism; concise on broader rationale.

2026.10.21-101049-3b2c077

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Mistral AI

Feedback ⓘ snapshot-20261022 NOTES ↗ RANK

Learn how to deploy and use Mistral AI's Large Language Models with our comprehensive documentation, guides, and tutorials.

> Mistral AI launched the Voxtral 3 family of speech-to-text and text-to-speech models plus a real-time voice agent SDK, while also shipping inline batch job creation, several smaller new models, and a round of breaking API changes and model deprecations.

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

01 **Breaking changes to Completions and Chat APIs** BREAKING

83

The Completion API now streams a new `p` parameter in each chunk, which may break strict chunk parsers — apps using `mistral-common` must upgrade to version 1.8.4 or higher. `tool_call_id` is now mandatory in chat messages with the `tool` role. The API also now strictly enforces parameter validity, returning the error `Extra inputs are not permitted` for unsupported parameters instead of silently ignoring them.

— Exact field, error string and required package version given. snapshot-20261022

02 **New model releases: Leanstral, Moderation 2603, Voxtral Mini Transcribe 2603**

NEW

63

Released `labs-leanstral-2603` (Leanstral), an open-source code agent for Lean 4 formal proof engineering; `mistral-moderation-2603` (Mistral Moderation 2603); and `voxtral-mini-2602` (Voxtral Mini Transcribe 2) plus `voxtral-mini-transcribe-realtime-2602` (Voxtral Mini Transcribe Realtime), both adding context biasing and `diarize` support in the Audio Transcriptions API.

— Named model ids and API support but no usage details. snapshot-20261022

THINNER COVERAGE BELOW

03 **Pro Voice speaker cloning adaptation** NEW

55

Pro Voice adaptation clones a speaker's voice from just 30 minutes of audio, reproducing their emotional range and speaking style across languages.

— Concrete audio requirement but no API or UI path. launch-20261020-26d56ceb

04 **Voice agent SDK for real-time pipelines** NEW

50

A new voice agent SDK combines real-time listening, reasoning, tool calling, and speaking into a single streaming pipeline, for phone and web deployments.

— Describes pipeline but no SDK name, method or install step. launch-20261020-26d56ceb

05 mistral-tiny renamed to mistral-tiny-2312 BREAKING

50

`mistral-tiny` is renamed to `mistral-tiny-2312`; the old endpoint will be deprecated in three months.

— Names old and new ids but no further migration mechanism. snapshot-20261022

06 Documentation URL restructuring under /vibe/ BREAKING

50

Documentation paths `/le-chat/*` and `/mistral-vibe/*` now redirect to the new `/vibe/*` tree.

— Exact old and new path prefixes named. snapshot-20261022

07 EU-hosted and self-deployed Voxtral 3 availability NEW

48

Voxtral 3 is available via Mistral's EU-hosted API and for self-deployment on private infrastructure, giving customers control over audio retention.

— Names deployment options but no config or endpoint details. launch-20261020-26d56ceb

08 Inline batch job creation without file upload NEW

35

Mistral AI's Batch API now supports inline batching, letting you create batch jobs without first uploading a file.

— No endpoint, flag, or format specified for the new batching mode. snapshot-20261022

└──▷ BREAKING ON UPGRADE

! A new parameter `p` added to chunks streamed by the Completion API may break applications relying on strict parsing of chunks; users relying on `mistral-common` package must update to version 1.8.4 or higher.

! `mistral-ocr-4-0` (OCR 4.0) is deprecated and retires on September 30, 2026; migrate to `mistral-ocr-4-1` or `mistral-ocr-latest`.

! `labs-leanstral-1-5` (Leanstral 1.5) is deprecated and retires on September 30, 2026.

! `zai-glm-5-2` (Z.ai GLM 5.2) is deprecated and retires on October 31, 2026; migrate to `zai-glm-5-3`.

! Documentation paths `/le-chat/*` and `/mistral-vibe/*` now redirect to the new `/vibe/*` tree.

! Devstral 2.0 moves to paid API access.

! `tool_call_id` is now mandatory in chat messages with the `tool` role.

! The API now strictly enforces validity of all parameters; requests with unsupported parameters return error `Extra inputs are not permitted` instead of silently ignoring them.

! `mistral-tiny-2312` is renamed from `mistral-tiny`; the old endpoint will be deprecated in three months.

WAS THIS USEFUL? 👍 👎

Anthropic

Feedback ⓘ snapshot-20261022 NOTES ↗ RANK

The Claude Developer Platform is the API for building applications on Anthropic's Claude models that generate, analyze, and automate work.

> Anthropic's biggest additions this window are sandboxed bash and text-editor code execution server tools, workspace-scoped rate limits/keys/reporting, and org-wide web fetch/search domain controls enforced in Claude Managed Agents, alongside a new API version (2026-10-21) shipped with breaking major SDK and CLI releases across every supported language.

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

01 Bash and text-editor code execution server tools NEW

95

Adds `bash_code_execution` server tool (type `server_tool_use`) that runs bash commands in a sandboxed container server-side, returning `stdout`, `stderr`, `return_code`, and captured files in `bash_code_execution_tool_result` blocks, streamed via `content_block_start` / `content_block_delta` SSE events with partial JSON for the `command` field. Adds `text_editor_code_execution` server tool supporting `view`, `create`, and `str_replace` commands, returning `text_editor_code_execution_view_result`, `text_editor_code_execution_create_result`, and `text_editor_code_execution_str_replace_result` blocks. Usage is reported in `usage.server_tool_use.code_execution_requests`; execution has a 5-minute minimum billed duration (billed even if the tool isn't called, when files are preloaded), and mixing this server-side tool with a client-provided shell tool creates two unsynced execution environments Claude may conflate.

— Full field and block names plus billing mechanics product docs

02 Workspace-scoped access, limits, and reporting NEW

90

Adds per-workspace rate limits (requests per minute, input tokens per minute, output tokens per minute) configurable under Rate limits in the Claude Console by selecting the target workspace; workspace-scoped API keys that restrict access to a single workspace; and a per-request workspace header letting non-scoped keys dynamically choose the workspace. Exposes `anthropic-workspace-id` and `anthropic-organization-id` response headers, workspace-level usage and cost reports, and the real workspace ID via the API key's `workspace_id` field, including for the Default Workspace.

— Names headers, config path and fields but no code sample product docs

03 1-hour cache TTL and caching behavior clarifications NEW

85

Adds `ephemeral_1h_input_tokens` field in the `cache_creation` usage response, enabling 1-hour TTL cache writes at 2x base input token cost alongside the existing 5-minute TTL. Clarifies that caches are isolated per workspace within an org on the Claude API, Claude Platform on AWS, and Microsoft Foundry, but only at the organization level on Bedrock. Documents that thinking blocks and server tool results are cached automatically during tool use — on Opus 4.5+, Sonnet 4.6+, and Haiku 5.5, thinking blocks stay cache-valid when non-tool-result user content is added, unlike earlier models — and adds guidance to pre-warm the cache with a startup or scheduled request.

— Names exact field, costs, and model versions `product docs`

---

04 Message Batches workspace isolation and caching discounts

IMPROVED

83

Batches are now scoped to the workspace they're created in, accessible only within that workspace or by Console users with permission to view workspace batches. Prompt-caching discounts can stack with the 50% Message Batches API discount, with cache hit rates ranging 30%-98% depending on traffic patterns. Orgs can disable batch-result downloads at the organization or per-workspace level via the Console; results remain downloadable for 29 days after batch creation, with the batch itself staying viewable after that window.

— Specific numbers and scope, no API field names given `product docs`

---

05 Org-wide web fetch and search domain controls NEW

83

Organization admins can disable web fetch entirely or configure allowed/blocked domains on the Capabilities page of the Claude Console; when disabled, Messages API requests including the web fetch tool fail with a `400` error. Claude Managed Agents now enforce these org-level web fetch and web search domain restrictions in addition to the per-tool `allowed_domains` / `blocked_domains` lists on the agent's `web_fetch` / `web_search` tools — a URL (including redirect destinations) must pass both lists, and blocked `web_fetch` calls return an error result to the agent instead of fetched content.

— Exact console path, status code, and tool fields named `snapshot-20261022`

---

06 API version 2026-10-21 and major SDK/CLI releases

BREAKING

80

Releases API version `2026-10-21` alongside major SDK versions that default to it — Python SDK `2.0.0`, TypeScript SDK `1.0.0`, C# SDK `13.0.0`, Go SDK `2.0.0`, Java SDK `3.0.0`, PHP SDK `1.0.0`, Ruby SDK `2.0.0`, and `ant` CLI `2.0.0` — each containing breaking changes independent of the API version bump. These versions

switch licensing from MIT to Apache 2.0, and callers relying on the previous default API version must review the migration guide.

— Exact version numbers for every SDK plus migration note `snapshot-20261022`

07 **Enforcement status fields on user profiles** NEW 65

Adds `enforcement_status` and `enforcement_actions` fields to user profiles so operators can see whether Anthropic has restricted or banned an end customer, plus an `enforcement_status[]` filter on the list-profiles endpoint to surface only profiles with an open restriction or ban.

— Names exact fields and filter, no endpoint path given `snapshot-20261022`

THINNER COVERAGE BELOW

08 **Per-request image limits for vision** IMPROVED 55

Supports up to 600 images per API request for models without a 200k-token context window, and up to 100 images per request for 200k-token models; claude.ai supports up to 20 images per message turn. Maximum image dimension is 8000×8000 px.

— Concrete numeric limits but no config or API surface `product docs`

09 **Mid-run steering for managed agents** NEW 35

Enables sending additional user events to a running Claude Managed Agent to guide it mid-execution or interrupt it to change direction.

— Described only in prose, no event schema or example `product docs`

└──▷ BREAKING ON UPGRADE

! The new Python SDK 2.0.0, TypeScript SDK 1.0.0, C# SDK 13.0.0, Go SDK 2.0.0, Java SDK 3.0.0, PHP SDK 1.0.0, Ruby SDK 2.0.0, and CLI 2.0.0 contain breaking changes independent of the API version change.

! The new SDK and CLI versions are licensed under Apache 2.0 instead of MIT, which may affect projects with license-compatibility requirements.

! The new SDK versions send API version `2026-10-21` by default; callers that relied on the previous default API version must review the migration guide.

! The new SDK and CLI major versions (Python `2.0.0`, TypeScript `1.0.0`, C# `13.0.0`, Go `2.0.0`, Java `3.0.0`, PHP `1.0.0`, Ruby `2.0.0`, `ant` CLI `2.0.0`) contain breaking changes independent of the API version bump.

! The new SDK and CLI versions are licensed under Apache 2.0 instead of the MIT License.

! When the org-level web fetch toggle is turned off, any Messages API request that includes the web fetch tool fails with a `400` error.

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

DeepSeek V4 Flash 0731 Feedback ⓘ

price increase · Oct 21, 2026 MODEL ↗

DeepSeek V4 Flash 0731 is a chat model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-v4-flash-0731): $0.016 → $0.44 in, $0.64 → $1.32 out per million tokens (+2650%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4 Pro 0813 Feedback ⓘ

price increase · Oct 21, 2026 MODEL ↗

DeepSeek V4 Pro 0813 is a chat model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-v4-pro): $0.2088 → $0.9553 in, $0.4176 → $1.9105 out per million tokens (+358%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4.1 Flash Feedback ⓘ

price increase · Oct 21, 2026 MODEL ↗

DeepSeek V4.1 Flash is a multimodal model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-v4.1-flash-1017): $0.15 → $0.30 in, $0.60 → $1.20 out per million tokens (+100%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.1 Feedback ⓘ price increase · Oct 21, 2026 MODEL ↗

GLM 5.1 is a chat model from Z.ai.

Price increase on OpenRouter (z-ai/glm-5.1): $0.966 → $1.40 in, $3.036 → $4.40 out per million tokens (+45%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.2 Feedback ⓘ price increase · Oct 21, 2026 MODEL ↗

GLM 5.2 is a chat model from Z.ai.

Price increase on OpenRouter (z-ai/glm-5.2): $0.62 → $0.0341 in, $2.42 → $7.00 out per million tokens (+189%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

Google Gemma 4 26B A4B Feedback ⓘ

price increase · Oct 21, 2026 MODEL ↗

Gemma 4 26B A4B is a multimodal model from Google.

Price increase on OpenRouter (google/gemma-4-26b-a4b-it): $0.0675 → $0.09 in, $0.225 → $0.30 out per million tokens (+33%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

NVIDIA Nemotron 3 Super

Feedback ⓘ

price increase · Oct 21, 2026 MODEL ↗

Nemotron 3 Super is a chat model from NVIDIA.

Price increase on OpenRouter (nvidia/nemotron-3-super-120b-a12b): $0.08 → $0.30 in, $0.45 → $2.40 out per million tokens (+433%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-122B-A10B

Feedback ⓘ

price increase · Oct 21, 2026 MODEL ↗

Qwen3.5-122B-A10B is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3.5-122b-a10b): $0.26 → $0.40 in, $2.08 → $3.20 out per million tokens (+54%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-9B

Feedback ⓘ price cut · Oct 21, 2026

MODEL ↗

Qwen3.5-9B is a multimodal model from Alibaba.

Price cut on OpenRouter (qwen/qwen3.5-9b): $0.10 → $0.08 in, $0.15 → $0.13 out per million tokens (-20%), first seen 21 Oct 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 22, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue