THE AI TOOLCHAIN

NO. 078

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 30, 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 884 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 151 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.

EVALUATE

Fail a build on a prompt regression instead of finding it in prod

vim ? help

`traccia eval` returns structured exit codes and gates on pass rate, cost and latency, per question if you want. The step that goes away is the human reading an eval table and deciding whether the number moved enough to care.

traccia-py

DEPLOY

Shrink a model without renting a second GPU for the teacher

HIGGS allocates bitwidths with an ILP rather than one uniform setting, and QADModifier distills block-wise using the unquantized weights as their own reference, so recovery no longer needs a separate teacher in memory. Strata adds Intel Arc B-series pipeline parallelism and a four-card Radeon profile for teams whose inference hardware is not NVIDIA.

LLM Compressor · Strata

BUILD

Post-train a trillion-parameter model on one node, and on AMD

v1.17.0 adds single-node fine-tuning for the 1T-parameter Mistral Large 4 and ROCm support, plus cross-tokenizer distillation in GOLDTrainer — teacher and student no longer have to share a vocabulary, which is usually what blocks distilling across model families.

TRL

GOVERN

Keep a stuck or wandering MCP server from poisoning the agent loop

yoagent withdraws tools when a stdio server stops responding, scopes HTTP redirects to their origin, and isolates stdio servers in their own process group so children get cleaned up. CHAP binds approval decisions to a digest, so what a human approved is what actually runs.

yoagent · chap

BUILD

◆ AI Coding Agents

Replit Agent

Feedback ⓘ changelog · Oct 30, 2026 CHANGELOG ↗

RANK

Replit Agent is an AI coding agent that builds, tests, and deploys applications in Replit.

> Replit Agent gained autonomous on-call monitoring with auto-fix and auto-republish, the ability to turn conversations into reusable skills, read-only chat/asset sharing, in-app Stripe Atlas incorporation, and a credit overage grace period for published apps.

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

01 **Autonomous on-call monitoring and auto-fix** NEW 80

Agent can act as an autonomous on-call engineer: enabling Monitoring alerts and Auto-fix lets it triage production alerts from logs and metrics in the background, and the experimental Auto-republish setting lets it push validated fixes automatically.

Enable background alert triage and automatic fix publishing so Agent handles production incidents without manual intervention.

> 📍 In your Replit project, go to Monitoring › Settings, turn on Monitoring alerts and Auto-fix, then also enable Auto-republish (experimental) to let Agent push validated fixes live automatically.

— Names exact settings and includes step-by-step enablement. changelog-20261030-67085935

02 **Chat and asset sharing with teammates** NEW 70

Share a whole chat or a single asset with selected teammates or the entire workspace; chat access is read-only and includes the chat's assets and development URLs.

Share a chat session and its assets with teammates so they can review Agent-built development URLs without needing edit access.

> 📍 Open the chat you want to share, select Share, choose specific teammates or Whole workspace, and confirm — recipients get read-only access including the chat's assets and development URLs.

— Exact UI steps given for sharing read-only chat access. changelog-20261030-67085935

THINNER COVERAGE BELOW

03 **Credit overage grace period for published apps** NEW 55

Eligible top-up customers can keep published apps running at zero credits up to an overage limit; paid Agent pauses while free mode stays available, and warning emails arrive before suspension.

— Explains eligibility and behavior but omits the actual overage limit.

changelog-20261030-67085935

04 **Agent-suggested workspace skills** NEW 50

Agent now suggests and applies updates to reusable workspace skills, turning useful conversation workflows into skills that Agent can propose or improve, pending your review before saving to the skills library.

— Describes mechanism but gives no exact UI path or config. changelog-20261030-67085935

05 **Stripe Atlas business incorporation** NEW 40

Ask Agent to incorporate your business through Stripe Atlas directly inside Replit, with 20% off when incorporating through Replit.

— Thin description with no steps beyond asking Agent. changelog-20261030-67085935

WAS THIS USEFUL? 👍 👎

Command Code

Feedback ⓘ changelog · Oct 30, 2026 CHANGELOG ↗

RANK

> Command Code's latest release adds a `/btw` side-question command, agent-driven artifact publishing, a new Providers settings page, mod support in the chat panel, BYOK model support in ACP, and a batch of CLI and UI refinements including cursor navigation shortcuts and live token speed metrics.

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

01 Mod support in chat panel NEW

70

Command Code mods — commands, notices, dialogs, status bar text, and skills — now work in the chat panel, extending where mods can run beyond the CLI.

Run an automated commit-and-push with a custom mod applied, letting you enforce team-specific commit conventions or hooks.

```
$ cmd acp --mod my-commit-mod
```

— Names mod types and has a runnable example command changelog-20261030-b69c409b

02 `/btw` side-question command NEW

65

`/btw` lets you ask a side question mid-turn without interrupting the main conversation, for example checking a quick fact while the agent is still working on its task.

Ask a quick clarifying question mid-session without losing the main thread — useful when you need to check a definition or side fact while the agent is mid-task.

```
$ /btw what does this error code mean?
```

— Named command with runnable example, little mechanism beyond that

changelog-20261030-b69c409b

03 Expanded CLI cursor navigation shortcuts NEW

65

The CLI input now supports `Ctrl+B` and `Ctrl+F` to move the cursor, and `Ctrl+D` or `Alt+D` to delete forward.

— Exact keybindings given, ready to use immediately changelog-20261030-b69c409b

04 Artifact publishing via `artifact_publish`/`artifact_manage`

NEW

65

HTML pages can now be published as shareable links directly from the agent using the `artifact_publish` and `artifact_manage` commands.

— Names both commands but no usage example shown changelog-20261030-b69c409b

THINNER COVERAGE BELOW

05 **Free-model background tasks with `/config` toggle** NEW

55

Background tasks now run on the active free model to save costs; this can be turned off with the `/config` toggle.

— Names the toggle but not its exact syntax changelog-20261030-b69c409b

06 **Providers page in Settings** NEW

55

Settings now includes a Providers page where you can add model providers and keys, or connect a ChatGPT or GitHub Copilot subscription.

— UI path described but no config keys or API details changelog-20261030-b69c409b

07 **Claude Code permission mode aliases for `--permission-mode`** NEW

50

Claude Code permission mode aliases can now be passed directly to the `--permission-mode` flag.

— Names the flag but gives no example values or mechanism changelog-20261030-b69c409b

08 **BYOK model support in ACP** NEW

40

Bring-your-own-key (BYOK) models are now supported and visible in ACP.

— States support exists but no setup steps given changelog-20261030-b69c409b

09 **Token speed metrics on responses** NEW

35

Finished responses now display time to first token and tokens per second.

— Clear what's shown but no location or config detail changelog-20261030-b69c409b

10 **Model picker follows live model catalog** IMPROVED

30

The model picker now follows the live model catalog so new models appear without requiring an extension update.

— Describes behavior change only, no names or steps changelog-20261030-b69c409b

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.2182 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 headline addition this window is universal web search plus image and video generation exposed to any agent via MCP, alongside faster account switching for Claude Code and Codex, DeepSeek peak-hour pricing awareness, and new usage tracking for MCP tool calls.

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

01 Tray account switching for Claude Code and Codex IMPROVED

85

Switch Claude Code or Codex accounts directly from the tray panel, which now reads the kept provider list via `GET /api/providers?last=1` so the menu is available immediately on open without waiting for a fresh provider fetch.

Query the cached provider list to confirm a Claude Code or Codex account switch is reflected immediately without triggering a full provider refresh.

```
$ curl 'http://localhost:<port>/api/providers?last=1'
```

— Exact endpoint with a runnable curl example v0.1.2174

02 Web search for any model via multiple providers NEW INTEGRATION

NEW

80

Add web search to any model that lacks it natively by routing through search providers (Tavily , Brave , Exa , Firecrawl , `You.com` , `DeepSeek` , Doubao Search on Volcengine, Yuanbao Search on Tencent Cloud, `SearXNG`); review usage in Settings or via `magpie search` . Control key rotation order per search provider with `pick=order` to cycle keys sequentially rather than by recency. Expose web search to any agent as an MCP server with a `web_search` tool and a `web_fetch` tool that retrieves a page by URL.

— Names providers, CLI command and flag but no exact config file path product docs

03 Add Claude account via OAuth setup token NEW

80

Add a Claude account using a setup token from `CLAUDE_CODE_OAUTH_TOKEN` via `AddClaudeSetupToken` — the token is stored in `logins.json` , named by a short hash (no email), and passed to Claude Code only for that account's own config directory runs, leaving any existing Claude Code sign-in untouched.

— Names env var and storage file, clear setup path v0.1.2167

04 **Usage logging for MCP tool calls** NEW 75

Track agent MCP tool calls in the request log with `usage.SourceMCP` rows — each row captures the tool name, provider/model, duration in milliseconds, and error message (if any), while omitting call arguments. Report tool calls answered by the Magpie Image server back to the gateway using the `magpie/toolCalled` JSON-RPC method (`MCPCalledMethod`) with `MCPCalled` params, enabling coverage of calls that never pass through the gateway directly.

— Named log schema and RPC method but no query example given v0.1.2170

05 **Peak-hour pricing model for DeepSeek** NEW 70

magpie now models DeepSeek peak-hour pricing with two schedules — `PeakDeepSeek` (01:00–04:00 and 06:00–10:00 UTC, Monday–Friday, excluding Chinese public holidays) and `PeakWeekdays` (same hours every weekday including holidays) — and applies a `PeakFactor` of 2× the off-peak list price during those windows.

— Precise schedule and factor named but not user-configurable v0.1.2173

06 **Scripts card groups HTTP-library clients in Context tab** IMPROVED 65

The Context tab now groups requests whose User-Agent identifies only an HTTP library, runtime, or SDK (such as curl, Bun, or python-httpx) under a single `Scripts` card, identified internally as `scriptsAgent` = `"scripts"`, listing the individual clients by frequency instead of giving each its own agent card.

— Named internal id but only a UI grouping change v0.1.2182

07 **Progressive rest ladder for out-of-credit API keys** IMPROVED 65

The gateway now applies a progressive rest ladder (`creditLadder`) to keys whose vendor reports an out-of-credit refusal — resting them for 2 hours, 8 hours, 24 hours, 3 days, and up to 7 days on successive refusals, instead of repeating the previous 30-minute cycle. The GUI now shows the date alongside the time for key rests that end on a future day, displaying a readable 'back at <date> <time>' instead of just a time-of-day.

— Exact timings given but no user-facing config to act on v0.1.2175

THINNER COVERAGE BELOW

08 **Image and video generation tools via MCP** NEW 50

Add an image generation tool for any agent over MCP, using a model of your choice, and a video generation tool backed by a Grok subscription or a MiniMax API key (MiniMax H3).

— Describes mechanism but no flag, endpoint or config key product docs

NEW INTEGRATION AnySearch MCP server

v0.1.2169 CODE ↗ STABLE

magpie now correctly treats a search API URL that already includes the vendor path as a base URL, preventing doubled path segments.

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.2169
```

git

```
$ git clone --branch v0.1.2169 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.2169
```

published under the project's GitHub account · checked 30 Oct Wrong command?

A search API URL given with the path magpie appends is now used as the base, so magpie no longer produces doubled paths like `.../search/search`.

v0.1.2162 CODE ↗ STABLE

Usage stats now correctly attribute Claude Desktop Code-tab sessions to Desktop rather than Claude Code, using entrypoint values to distinguish them.

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.2162
```

git

```
$ git clone --branch v0.1.2162 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.2162
```

published under the project's GitHub account · checked 30 Oct Wrong command?

WAS THIS USEFUL? 👍 👎

mvschwarz OpenRig

Feedback ⓘ v0.7.6 NOTES ↗ RANK

OpenRig is a network manager for AI coding agents: build persistent teams with roles, shared context and owned work.

> OpenRig's two releases add live-following terminal views for slices and tasks, in-conversation account switching for running seats, a new viewer-ensure API endpoint, and several breaking changes affecting daemon startup, escalation recovery, and API error codes.

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

01 **Live terminal workspaces and pane status** NEW 95

Adds live-following terminal workspaces for slices and tasks via `rig terminal open slice:<name-or-id> --follow --provider herdr` and `rig terminal open qitem:<id> --follow --provider herdr`, showing each owner's pending, in-progress, and blocked queue rows; tag rows with `rig queue create --slice <name-or-id>`, with handoffs keeping the tag, and tune behavior with `--pin`, `--suppress`, `--max-tiles`, and `--no-follow` (defaults: eight tiles, four seconds of departure grace, and twelve more seconds for a focused leaver). Terminal results now also distinguish `pane_unconfirmed` from opened and skipped panes, and an all-unconfirmed open exits nonzero.

Watch who is working a named slice right now — tiles appear and disappear as work is handed off between agents.

```
$ rig terminal open slice:auth-refactor --follow --provider herdr
```

Watch which agent currently owns a task in real time, with the view capped to four tiles and a pinned seat always visible.

```
$ rig terminal open qitem:abc123 --follow --provider herdr --max-tiles 4 --pin dev.impl
```

— Names exact commands, flags and numeric defaults States new status and exit behaviour, no example

v0.7.6 v0.7.5

02 **Live account-switching for running seats** NEW 95

Move a running seat to another account home with `rig provider switch --seat <seat> --to-account <name> --account-home <dir>`, previewing changes first with `--dry-run`. A Claude seat can switch mid-conversation between subscription

and API key using `--to-account api --api-key-file <file>` or back with `--to-account plan --token-file <file>`, and a Claude member's token file can be named directly in its spec with the `claude_token` field.

Move a running seat to a different account home without losing its address or in-flight work, verifying the change before applying it.

```
$ rig provider switch --seat impl --to-account personal --
account-home ~/accounts/personal --dry-run
```

— Runnable command with dry-run flag and config field named v0.7.6

03 **Viewer-ensure endpoint for terminal panes** BREAKING

83

`POST /api/terminal/viewer/ensure` adds or reuses a verified viewer without rearranging existing panes; membership grants no input authority. As of v0.7.6 it returns HTTP 400 (previously HTTP 200 with an error body) for `view_required` and `unknown_provider` error codes, so callers must handle the 400 status while still reading the body's code field.

— Named endpoint and exact status-code change given v0.7.6 v0.7.5

04 **Held-message handling for Codex seats** IMPROVED

80

Check messages held before delivery to a Codex seat with `rig seat held-messages`. With a Codex seat's typing guard set to `draft-aware` or `openrig-urgent`, a message OpenRig cannot confirm now stays held before it is typed, rather than being left typed and unsent in the composer.

— Named command plus before/after behaviour change v0.7.5 v0.7.6

05 **Runner-following and model passing for community-harness presets**

NEW

70

Enable community-harness presets to follow their runner's activity with `runner: true` (requires `launch_result: true`), and receive the declared model as `OPENRIG_MODEL` with `passes_model: true`.

— Config keys named but no worked example v0.7.6

06 **Disable Slack reaction acknowledgements** NEW

60

Disable acknowledgement emoji from waking agents in Slack with `rig slack setup --reactions off`.

Stop Slack reaction emoji from waking agents on every acknowledgement, reducing unwanted interruptions in busy channels.

```
$ rig slack setup --reactions off
```

— Single runnable command, thin on mechanism v0.7.6

---

07 **Daemon start readiness-deadline handling** BREAKING 60

When only the readiness deadline passes during startup, `rig daemon start` now exits nonzero with a pending message instead of killing the daemon; `rig daemon stop` cannot reach the unpublished process, and the printed message must be followed for manual recovery.

— Describes new failure mode but no recovery command v0.7.6

THINNER COVERAGE BELOW

08 **Escalation edges affect seat recovery** BREAKING 45

Previously inert `escalates_to` edges now affect recovery for seats without a delegation parent after a database migration runs on upgrade.

— Mechanism named, no migration steps given v0.7.5

---

09 **Runtime send refusal status code change** BREAKING 35

Runtime send refusals now return HTTP 409 instead of 500.

— Single status-code fact, no further detail v0.7.5

└──▷ **BREAKING ON UPGRADE**

! `POST /api/terminal/viewer/ensure` now returns HTTP 400 (instead of HTTP 200 with an error body) for `view_required` and `unknown_provider` error codes; callers must handle the 400 status while still reading the body's code field.

! When only the readiness deadline passes during startup, `rig daemon start` now exits nonzero with a pending message instead of killing the daemon; `rig daemon stop` cannot reach the unpublished process, and the printed message must be followed for manual recovery.

! Previously inert `escalates_to` edges now affect recovery for seats without a delegation parent after a database migration runs on upgrade.

! Runtime send refusals now return HTTP 409 instead of 500.

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.34.32 NOTES ↗ RANK

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

> Letta Code's CLI gained a full-body agent creation flag and new scripting capabilities for CodeMode, Monitor, and Workflow, alongside environment-variable support for commit trailers and sandboxed scripts.

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

01 `agents create --body` flag for full agent specs NEW

80

The `agents create` command gains a `--body` flag that accepts a complete JSON request body, letting a user fully specify an agent's config (name, model, system prompt, etc.) in a single CLI call instead of assembling it from individual flags.

Create an agent with a fully specified configuration in one shot, rather than assembling it from individual flags.

```
$ letta agents create --body '{"name": "my-agent", "model": "gpt-4o", "system": "You are a helpful assistant."}'
```

— Runnable command with exact flag shown v0.34.32

02 `BASH_COMMIT_TRAILER` env var for commit trailers NEW

75

Setting the `BASH_COMMIT_TRAILER` environment variable adds a custom git commit trailer to all agent-authored commits, useful for audit trails or CI filtering.

Stamp every commit the agent makes with a team-specific trailer (e.g. for audit trails or CI filtering) by exporting the variable before running Letta Code.

```
$ export BASH_COMMIT_TRAILER='Reviewed-by: letta-agent'
letta
```

— Named env var with runnable export command v0.34.27

THINNER COVERAGE BELOW

03 `process.env` access in CodeMode and Workflow scripts NEW

55

CodeMode and Workflow scripts can now read environment variables at runtime via `process.env`.

— Names the surface but no sample usage given v0.34.29

04 **Turn-vs-compaction signal to local model endpoints** NEW 40

Local model endpoints now receive a signal indicating whether a given request is a normal turn or a compaction, allowing endpoint-side routing or behavior differences based on request type.

— Explains purpose but no named field or config v0.34.32

05 **Server-side tool calls from CodeMode, Monitor, and Workflow** NEW 40

CodeMode, Monitor, and Workflow scripts can now call server-side tools directly.

— Names three surfaces but no mechanism or example v0.34.28

06 **Video-generation skill in agent library** NEW 35

A video-generation skill, adapted from OpenRouter's implementation, has been added to the agent skill library.

— Names source but no usage detail v0.34.27

07 **Configurable sandbox monitor duration cap** IMPROVED 30

The sandbox monitor's duration cap is now configurable, allowing longer or shorter monitoring windows to be set per deployment.

— No config key or flag named v0.34.29

08 **Model discovery for external coding workers** NEW 30

Letta Code now discovers models for external coding workers, expanding which backends can be assigned coding tasks.

— No command or config surface named v0.34.27

09 **Isolated product credential stores** NEW 30

Letta Code now supports isolated product credential stores, keeping credentials for different products separate.

— No config key or API named v0.34.27

WAS THIS USEFUL?  

The Open Engine Zeroshot

Feedback ⓘ v10.21.0 NOTES ↗

RANK

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

> Zeroshot's v10.21.0 release adds a planning skill that produces structured plan files, a TypeSafe-compatible decision adapter for verifier nodes, GitHub Enterprise and GitLab support for PR/MR workflows, and caller-declared model context limits.

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

01 **GitHub Enterprise and GitLab host support** NEW 74

Zeroshot now supports pull/merge request creation, merge delivery, and issue/review comments on publicly reachable GitHub Enterprise (Server 3.22.2+) and GitLab (19.5.1+) instances. Local runs reuse host-scoped `gh` or `glab` logins, while hosted runs accept explicit token connections.

— Names versions and auth mechanisms, but no direct command shown. v10.21.0

02 **Caller-declared model context limits** NEW 72

Agent runtime bindings can now declare optional `contextWindow` and `maxTokens` fields so callers state a model's limits instead of Zeroshot guessing them; absent values keep each provider's defaults.

Pin a model's context and output limits in a runtime config so Zeroshot uses your values instead of provider defaults.

```json
{
  "harness": "codex",
  "provider": "openai",
  "model": "YOUR_MODEL_ID",
  "effort": "high",
  "contextWindow": 128000,
  "maxTokens": 4096
}
```

— Config fields named and shown in a runnable example. v10.21.0

03 **TypeSafe-compatible decision runtime binding** NEW 70

A new `decision` runtime binding lets verifier nodes call a model through a TypeSafe-compatible System One endpoint using a caller-provided endpoint and API key. It

validates the complete probability distribution and emits uncertainty signals for ties or results below the chosen threshold.

— Describes mechanism and signals but no concrete config example given. v10.21.0

---

04 **zeroshot-plan planning skill** NEW 67

A new `zeroshot-plan` skill turns a request or issue into a plan file at `.zeroshot/plans/<id>.md` (git-ignored) with checkable acceptance criteria, asking only questions that change the outcome.

Turn a vague issue into a structured plan file before any code is written, so acceptance criteria are agreed up front.

> 📍 Use the zeroshot-plan skill to plan this issue before building anything.

— Names exact file path and behaviour, with a usage example. v10.21.0

THINNER COVERAGE BELOW

05 **Removal of --no-remote-export flag** BREAKING 55

The `--no-remote-export` flag has been removed; use `--sync-session off` instead for current Copilot CLI releases.

— Exact flags named with migration path, minimal context otherwise. v10.21.0

└──▷ **BREAKING ON UPGRADE**

! The `--no-remote-export` flag has been removed; use `--sync-session off` instead for current Copilot CLI releases.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.314 NOTES ↗

CODE ↗ RANK

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

> Claude Code's v2.1.314 release adds directory-scoped write enforcement, IdP group-based routing for gateway upstreams, Bedrock guardrail trace logging, and automatic summarization of long idle conversations in VS Code and Claude Desktop.

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

01 Write-path restriction to working directories NEW

85

New `permissions.blockWritesOutsideWorkingDirectories` config setting causes the Write, Edit and NotebookEdit tools to refuse any file path outside the configured working directories, and this enforcement applies in every permission mode.

Prevent Claude from editing files outside your project root — useful in multi-repo workspaces or when running untrusted prompts.

```json
{
  "permissions": {
    "blockWritesOutsideWorkingDirectories": true
  }
}
```

— Exact config key and runnable JSON example given v2.1.314

02 IdP group routing for gateway upstreams NEW

75

An optional `groups` list can now be added to Claude apps gateway upstream configuration so that only developers in the listed IdP groups are routed to that upstream, including during failover.

Route only members of your `security-team` IdP group to a hardened gateway upstream, falling back to the same rule on failover.

```yaml
upstreams:
  - url: https://secure-gateway.example.com
    groups:
      - security-team
```

— Named config field with YAML example, mechanism noted v2.1.314

03 **Bedrock guardrail trace logging** NEW 65

Adding `guardrail_trace` to Claude apps gateway Bedrock upstreams logs which Amazon Bedrock guardrail filter fired on a request, without logging the matched text itself.

— Named config field but no usage example provided v2.1.314

THINNER COVERAGE BELOW

04 **Auto-summarize idle long conversations** IMPROVED 55

VS Code and Claude Desktop's Code tab now automatically summarize conversations of 200K+ tokens that have been left idle, and use that summary on the next message once the prompt cache has expired.

— Describes behavior and trigger but no user-facing control v2.1.314

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ desktop-v2.0.2 NOTES ↗ RANK

Superset is a desktop workspace for running CLI coding agents like Claude Code with terminals, diff viewer, and browser previews.

> Superset v2.0.2 ships viewer-identity PR reviews, a new right-pane workspace area, and Devs — persistent background agents with their own activity tracking and failure diagnostics — alongside smaller improvements to agent setup, chat notifications, and mobile layout.

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

01 **Devs: persistent background agents** NEW 75

Superset adds Devs — persistent agents with one main session per Dev, tasks running on their own boxes, background work and PRs surfaced in the top bar, a Dev doctor that diagnoses failures, and an Activity tab showing live working state. Each turn runs via an agent worker using `pdev`, hosted on Cloudflare, with a Dev's main session running on its own model.

— Explains mechanism (pdev, Cloudflare) and multiple concrete surfaces desktop-v2.0.2

02 **Viewer-identity PR reviews** NEW 70

Superset's PR review experience now acts as the viewer's own GitHub login (never the host's or the org's GitHub App) when reading and reviewing pull requests, adds review threads in the diff pane and Review tab shown under the viewer's own identity, and adds PR badges in chat that open the PR pane directly.

— Mechanism and UI surfaces named, but no exact nav steps desktop-v2.0.2

03 **Right-pane workspace area** NEW 60

Superset adds a right-pane area, now enabled for all users, with a New Tab screen, file tabs with preview and pinned states, tools accessible from the main `+` menu, and a file tree with a placeholder for an empty Files pane.

— Names concrete UI elements but no deeper mechanism desktop-v2.0.2

THINNER COVERAGE BELOW

04 **Agent setup and workspace creation improvements** IMPROVED 40

Superset adds agent-led environment setup, project links, a new-workspace row that remembers previous choices, Antigravity status reporting, and a model picker in agent setup.

— Lists several thin additions without detail on each desktop-v2.0.2

05 **Channel join/leave messages and notifications** NEW

35

Superset adds channel join and leave messages (as in Slack) and now notifies on all new channel messages by default.

— Describes behaviour but no config or opt-out mentioned `desktop-v2.0.2`

06 **Mobile layout support for foldable and tablet devices**

IMPROVED

30

Superset's mobile layout now adapts for iPhone Duo's unfolded display and iPad.

— Names devices but no detail on what changed `desktop-v2.0.2`

07 **Fork agent session between CLI and chat** NEW

20

Superset adds the ability to fork an agent session between CLI and chat.

— Bare description with no mechanism or usage path `desktop-v2.0.2`

WAS THIS USEFUL?  

codealta

Feedback ⓘ 3.1.0 NOTES ↗ RANK

Agentic Development Environment

> CodeAlta's biggest window of the year adds headless, sandboxed agent execution via `altacli`, agent-authored JavaScript workflows, a PR-style code review gate for commits, full mobile session control over Tailscale, and a wave of new provider integrations (ACP registry, Codex CLI, llama.cpp, Mistral) alongside MCP server skills, local full-text/semantic search, project labels, and numerous worktree and editor-integration improvements.

└──▷ **WHAT SHIPPED · 46 FEATURES** most completely described first ? what's the number?

01 **Headless agent runs via altacli** NEW 95

Run agents without a window using `altacli run`, with `--format jsonl` for an ACP-format JSON Lines trace, `--output-schema` for schema-validated JSON output, `--sandbox workspace` or `--sandbox strict` for network profiles, `--timeout` for a time limit, `--profile`, `--isolated` and `--ephemeral` to keep runs apart from your sessions, and exit codes that report how the run ended. Serve CodeAlta as an ACP agent over stdio with `altacli acp` (including the MCP servers the editor provides), run `alta` agent commands from a script with `altacli alta`, and control prompt parts sent to the agent with `--enable` and `--disable` flags on `altacli`.

Run a headless agent task in CI and capture a full ACP-format event trace with a workspace network profile and a 10-minute timeout.

```
$ altacli run --format jsonl --sandbox workspace --timeout 600 -
< task.md > events.jsonl
```

Review a diff with a specific model without opening any UI, piping git output straight to the agent.

```
$ git diff | altacli run -m anthropic:claude-opus-5-5@high
"Review this diff"
```

Let a CI bot start CodeAlta Desktop to run automations and have it exit automatically once all sessions finish, with no interactive prompts.

```
$ alta --automated
```

— Multiple runnable commands with exact flags given. 3.0.0 2.1.0

02 **Mobile session control over Tailscale** NEW 87

Enabling the built-in Mobile plugin in Settings › Plugins, opening the Mobile tab with Ctrl+G M, and scanning its QR code lets you control sessions from a phone — steering running turns, queuing prompts, stopping agents, and changing model or reasoning effort; the page connects through Tailscale on your own tailnet and requires the access token embedded in the QR code. The mobile page can now be filtered by one or more projects (checked projects shown first), and you can attach files and pictures, answer agent questions and review requests, and see workflow cards with sub-agent listings from mobile.

— Exact shortcut and settings path given, no API surface. 3.0.0 2.1.0 2.0.0

03 **MCP server skills: activation, approval and invocation** NEW

86

Activate skills published by MCP servers using `alta mcp skill activate`, with per-skill approval prompts showing `SKILL.md` and Allow/Allow once/Deny options, managed in Settings › Skills. List and inspect skills with `alta mcp skill list`, `alta mcp skill show`, and `alta mcp skill read`. Skills can also be searched and invoked directly from the prompt by typing `/` or opening Ctrl+P and selecting the Skills category, using `/name <context>` to activate a specific skill with additional instructions.

Activate an MCP server skill for use in a session after reviewing its SKILL.md.

```
$ alta mcp skill activate transformers-js
```

List all skills published by connected MCP servers to see what is available for approval.

```
$ alta mcp skill list
```

— Multiple runnable CLI commands and a slash syntax given. 1.12.0 1.9.3

04 **Project labels across Explorer, CLI and Settings** NEW

85

CodeAlta adds a label system for projects: each label carries a name, icon, and color, labels can be nested (e.g. Parsers › Config formats), and a project can carry multiple labels and appears under each in the Explorer. List all projects carrying a given label with `alta project list --label <name>`, create and assign labels from the CLI with the new `alta label` command, and browse and edit the full label tree under the new Settings › Labels page.

Scope an agent run to every project in a given category, such as all infrastructure repos.

```
$ alta project list --label infra
```

— Runnable CLI example and full UI path given. 2.2.0

05 **PR-style code review workflow with agent gating** NEW

82

Enable per-session or global review gating in `Settings › Sessions` with the `Review before commit` toggle, or per session from the Agent chip under the prompt: agents call `alta review` to request review, commit only on approval, and CodeAlta blocks any other commit regardless of permission mode. Use `alta review request file` to review a file or uncommitted changes, comment line-by-line on a plan, file, or uncommitted changes, attach a verdict (Approve, Request changes, or Comment), and submit via `Submit review`; pending reviews appear in Work items › Reviews.

Gate all agent commits behind your approval for a session — the agent stops and waits after proposing changes, committing only what you explicitly approve.

```
$ alta review request file
```

— Runnable command and full review flow given. `1.11.0`

06 **New provider integrations** NEW 80

Adds ACP registry agents (including Cursor, Gemini CLI, Qwen Code, and the Copilot CLI) as providers via `Settings › Providers › Add provider › ACP registry...`, downloaded and run through `npx` or `uvx` with their tools, permissions, and session history exposed alongside other providers; adds the Codex CLI as a provider via `Settings › Providers › Add provider › Codex CLI...`, accepting ChatGPT Business and Enterprise sign-in where other apps are refused, and generating images when used with a model that supports image reading, with the result shown in the session and a copy kept alongside it; adds a `llama-cpp` provider type that connects to a local or networked `llama-server` with no API key required, automatic context detection for compaction, image input for vision-projector models, and its own `api = "responses"` support for the llama.cpp server's Responses API; and adds Mistral's `api = "responses"` support (preview, with Chat Completions remaining the default) plus a new Mistral Vibe provider that signs in via browser using a Mistral Pro subscription, reachable at `Settings › Providers › Add provider › Mistral Vibe`.

Run a llama.cpp server locally and add it as a provider in CodeAlta without an API key, using its Responses API.

```text
# In your CodeAlta provider configuration:
# Provider type: llama-cpp
# api = "responses"
```

Switch the Mistral provider to the preview Responses API instead of the default Chat Completions API.

```toml
api = "responses"
```

— Exact UI path and runtime named, no CLI command. UI path and sign-in types named, no flags. Config snippet given, mechanism well described. Config key and UI path given, thin overall.

2.0.0 3.1.0 3.0.0

---

07 **On-demand MCP tool loading via tool search** NEW 80

On-demand MCP tool loading across CLI and API: `alta search` searches conversations, plans, skills, and MCP tools (with `alta search load` to load a single tool without its server), and on the Claude API (Claude 4.5 and later), MCP server tools are sent via `tool_search` without filling the context so the model loads only what it needs.

Have an agent find a relevant earlier conversation or skill without bloating the prompt context.

```
$ alta search "where did we fix the scroll jump"
```

— Command, API behavior and model version all named. 1.10.0

---

08 **Local full-text and semantic search** NEW 78

Local semantic search (search by meaning) uses on-device models EmbeddingGemma 2 (304 MB) or Granite (117 MB), indexed in the background, with a new Search canvas at `/search_index` to inspect what is indexed, compare keyword and semantic results side by side, and review search usage. Press Ctrl+P to search Conversations, Docs, and Tools from the same local index alongside sessions, projects, and files.

— Exact models, sizes, path and shortcut given. 1.10.0

---

09 **Per-prompt model and permission control via CLI** NEW 75

Pass `--model`, `--reasoning`, and `--permission-mode` to `alta session send` and `alta session queue` to set the model, reasoning level, and permission mode per prompt from the CLI; list what model and permission choices are available for a session with the new `alta session choices` subcommand.

Check which models and permission modes a session can run with before sending a prompt.

```
$ alta session choices
```

— Runnable command and named flags both given. 2.0.0

10 **Sandboxed sessions built on MXC** NEW 75

Run agent commands and file tools inside an OS container built on MXC, configured in Settings › Sandbox per session, provider, model, space or project, with `workspace` and `strict` network profiles; agents can read sandbox profile details and a session's active sandbox with `alta sandbox`, which reports what the profile allows. Claude Code providers running inside the sandbox now also expose an environment-variable surface.

— Command and settings path given, no full walkthrough. 2.2.1 2.1.0

11 **Agent-authored workflows via alta workflow/script** NEW 72

Run agent-authored JavaScript workflows with `alta workflow` and `alta script` CLI commands, which plan, dispatch sub-agents in parallel, sleep, and return Markdown results drawn live as a graph in the timeline.

— Two named CLI commands, no invocation example. 3.0.0

12 **External editor and file manager integration** NEW 72

Open your preferred editor or file manager from the prompt with the new `/external_editor` and `/file_explorer` commands. Choose any editor — including Visual Studio Code, Insiders, Cursor, Windsurf, Zed, or Sublime Text — or any program with custom arguments via `Settings > General > External editor`. Open in Visual Studio Code and Open in File Explorer (Finder on macOS) actions are now available in the project menu, the Workspace popover path, the Changes tab diffs, the Worktrees window rows, and the code editor menus.

Jump straight from the AI prompt to your preferred editor on the current session folder without leaving the conversation.

```
$ /external_editor
```

Browse the session folder in the OS file manager to inspect generated files or drag them elsewhere.

```
$ /file_explorer
```

— Slash commands and exact settings path both given. 1.9.2

13 **alta ask no longer accepts a file argument** BREAKING

67

`alta review` now handles file and change review; `alta ask` no longer accepts a `file` argument. Scripts or prompts that passed `file` to `alta ask` now receive `ask.fileMoved` and must switch to `alta review request file`. A custom `default.prompt.md` that replaces the built-in prompt must also retain its Review by the user section to remain compatible with the new review workflow.

— Exact error code and migration command given. 1.11.0

14 **File, folder and issue attachments with document conversion** NEW

67

Attach files or photos (Ctrl+U), a folder (Ctrl+Alt+F), or issues (Ctrl+Alt+I) to a prompt from the prompt bar's paperclip; Word, Excel, PowerPoint, EPUB, and RTF documents are converted to Markdown for any model.

— Exact shortcuts and supported formats named. 1.10.0

15 **Provider and model configuration enhancements** NEW

65

Enable image input on custom models by setting `supports_attachments` in a model's overrides; control the sort order of a provider's model list (provider order, by name, or by name and version) from the provider's configuration form; and add `${session.id}` , `${run.id}` , `${model.id}` and `${provider.key}` placeholders to the request headers of an `openai-chat` or `openai-responses` provider — for example the `x-opencode-session` header that OpenCode Go and Zen require.

— Named config key but no file location or example. Thin description, no config key named. Exact placeholder names and example header given.

3.1.0 2.1.0 1.11.0

16 **Claude Code session sharing with VS Code and terminal** NEW

65

Claude Code conversations started in VS Code or with `claude` in a terminal now appear in the Explorer under From Claude Code ; continue them in CodeAlta, open a CodeAlta session in VS Code via Open in VS Code , run it in a terminal via Run in terminal , or copy a resume command via Copy resume command .

— Named UI actions, no exact resume command text. 1.11.0

17 **Add projects from the terminal with `alta .`** NEW 65

Add projects from the terminal with `alta .` to add the current folder or `alta <folder>…` to add several at once, with `--space` to place them in a space.

Add your current working directory as a CodeAlta project from any terminal, placing it in a named space.



— Runnable command with flag and example given. `1.10.0`

---

18 **Named Azure DevOps issue views** NEW 64

Save and share Azure DevOps issue filters as named views in `.alta/issues.toml`, then switch between them beside the project or let agents use them with `alta issue`.

— Config file path and CLI command both named. `1.12.0`

---

19 **Session behavior settings: prompt suggestions and commit authorship**

IMPROVED 62

Prompt suggestions — where the model pre-fills the next prompt after each turn, accepted with `Tab` or `→` — now work with every provider, not only Claude Code, toggled in `Settings › Sessions › Prompt suggestions`; whether agents append a `Co-authored-by: CodeAlta` line to commits is controlled in `Settings › Sessions › Commits`.

— Exact settings paths given for both toggles. `2.0.0`

---

20 **Plugin system enhancements for canvas scripts and tool-call hooks**

NEW 62

Plugin canvas scripts can now import `@xyflow/react` to draw panning, zooming and draggable graphs such as pipelines, dependency maps and architecture diagrams, with a `canvas-flow` sample included; and plugin hooks can now inspect, block, or modify every tool call (including Code Mode script calls and MCP tool calls) before or after execution, and continue a session when its run ends.

— Package and sample named, no API reference. Mechanism described, no concrete hook API name.

`3.1.0` `3.0.0`

21 **Custom speech models for Speech settings** NEW 60

Use SenseVoice, Paraformer, or FireRedASR as a custom speech model on the Speech settings page, including multilingual SenseVoice covering Chinese, English, Japanese, Korean, and Cantonese.

— Named models and languages, no setup steps. 2.0.0

---

22 **present() for live Code Mode script output** NEW 60

Call present(markdown) in a Code Mode script to stream progress, findings and reports to the session card in real time as the script runs, without the agent repeating them.

— Function named, no full script example given. 3.1.0

---

23 **Timeline image folding and copying** NEW 60

A focused timeline collapses agent work between messages into one summary line — thoughts, tool calls, failures, lines changed and duration — enabled via `Settings > Appearance > Focused timeline` (on by default; turn it off to restore the expanded view), with folded timeline lines now also showing images produced during that work; copy a screenshot of the timeline to the clipboard by right-clicking a thumbnail or pressing Shift+F10 and choosing Copy image, or using Ctrl+C in the image window.

— Settings path given, default state noted. Exact shortcuts and menu action given. 1.11.2 1.9.3

---

24 **Composed agent prompts via inherit key** NEW 60

Compose agent prompts from shared base prompts with the `inherit:` key (for example `inherit: [default, maui]`) in a prompt definition.

Compose a reusable base prompt and have agent prompts inherit it so shared rules live in one place.

```yaml
inherit: [default, maui]
```

— Config example with exact key and syntax given. 1.10.0

THINNER COVERAGE BELOW

25 **Automatic session resumption on rate-limit reset** NEW

56

Turn on automatic session resumption via `Settings › Models` under the fallback with `Resume when the limit resets`, so sessions continue by themselves once the earliest tier limit resets.

– Exact toggle and settings path given. 3.0.0

26 **MCP server networking controls** IMPROVED

56

Add host names to `--mcp-host` so clients that refuse an IP address can connect to the MCP server, and add a Tailscale network switch to the CodeAlta MCP server settings to control Tailscale connectivity per server.

– Flag and settings switch named, no example. 2.2.1

27 **Find text in rendered content** NEW

56

Find text in rendered Markdown pages, plans, files under review, and Work items using `Ctrl+F`, with `Alt+C`, `Alt+W`, and `Alt+R` toggles for case, whole-word, and regex matching.

– Exact shortcuts for find and match modes given. 1.12.0

28 **Skill and MCP tool usage statistics** NEW

56

View per-skill usage statistics from the CLI with `alta statistics skills` and per-MCP-tool call counts with `alta statistics mcp`; the Tools page of Statistics now lists every skill and MCP server used in a period, including call counts, source, and last-used date.

– Named CLI commands, no example output shown. 1.10.0

29 **Faster and smarter worktree creation** IMPROVED

55

Worktrees are now prepared in the background while you are still typing the prompt for a new session, so the checkout is ready when you send, and worktree files are checked out in parallel, reducing wait time when creating multiple worktrees. New worktrees for repositories with submodules get those submodules checked out automatically, with progress shown on the Starting tab (if one submodule cannot be fetched, the session still starts and reports the failure), and the Workspace chip for a worktree session now opens the code editor on that worktree directly, with the `+` / `−` change counts opening the Changes view in one click.

– Behavioral improvements described, no config surface. 1.9.4 1.9.2

30 **Code Mode per-provider configuration and batching** IMPROVED

53

Enable or configure Code Mode per provider and model via `Settings › Code Mode`; it is on by default and agents now batch reads and edits to reduce turns and tokens.

— Settings path given, no further config keys. 3.0.0

31 **Per-model catalog details on Models page** NEW 52

See per-model prices, input types, knowledge cutoff, and release date on the Models page, matched to the models.dev catalog by endpoint and vendor.

— Describes the page, no interaction detail. 2.0.0

32 **Multi-window tab management** NEW 50

Use the `alta window` command group to let agents arrange tabs across monitors, for example placing a session's changes into a named Review window on a second monitor.

— Command named, example scenario but no syntax. 1.11.0

33 **Web search for ChatGPT/Codex subscriptions** NEW 48

Web search is now available for ChatGPT/Codex subscriptions, configurable as `cached`, `live`, or `disabled`.

— Named config options, no toggle location given. 1.10.0

34 **Session compatibility impact of tool search upgrade** BREAKING 48

After upgrading, each existing session restarts its tools once because the tool list has changed; on Anthropic, the first tool loaded by tool search costs one prompt-cache fill. A session that loaded tools via tool search cannot be opened by an older version of CodeAlta.

— Impact explained, no mitigation steps offered. 1.10.0

35 **MCP Apps view rendering in conversation** NEW 47

MCP servers that follow the MCP Apps extension now render their view (charts, forms, dashboards) directly in the conversation in CodeAlta Desktop, under the turn that called their tool.

— Describes behavior, no enabling steps given. 1.12.0

36 **Cross-project sub-session linking in Explorer** IMPROVED 40

Cross-project sub-sessions are now linked to their parent in the Explorer: an arrow under the parent opens the child, a line under the child opens the parent, and hovering a

session highlights its parent and children.

— UI behavior described, no named surface. 3.1.0

37 **Inline image rendering in Markdown messages** IMPROVED

40

Inline images referenced in Markdown messages and workflow results (![chart](chart.png)) now render in the timeline with copy and reveal actions.

— Markdown syntax shown, no toggle mentioned. 3.0.0

38 **Sessions without sub-sessions option** NEW

39

Create sessions that do all work themselves, without tools that start sub-sessions, using the new Sessions without sub-sessions option.

— Option named but no location given. 3.0.0

39 **macOS .NET runtime version warning** NEW

38

CodeAlta now warns macOS users when the installed .NET runtime is older than 10.0.13, surfacing the requirement before it causes silent failures.

— Exact version threshold given, no remediation step. 1.10.1

40 **Keep computer awake setting** NEW

36

A new Keep computer awake setting in Settings › General prevents the system from sleeping while CodeAlta runs, keeping automations and reminders active.

— Settings path given, thin single toggle. 1.10.0

41 **Antigravity streaming turns** IMPROVED

35

Antigravity turns now stream tool calls and text as they arrive rather than displaying them only at the end of the turn.

— Behavior change only, no configuration surface. 2.1.0

42 **CLI profiles tied to space identity** NEW

33

CodeAlta now associates CLI profiles with the identity of a space, letting each space carry its own command-line credentials alongside its other settings.

— Thin description, no command or key shown. 1.10.1

43 **Per-feature debug log levels** NEW

32

Set per-feature debug log levels via Settings (for example, scoping Debug to the issue-tracker feature while others stay quiet).

— Example given but no exact setting names. 3.0.0

44 **Bulk session start from Issues and Work Items tabs** NEW

32

The Issues and Work Items tabs now support starting several sessions at once.

— One-line UI addition, no further detail. `1.9.4`

---

45 **Composer Agent chip in Automations, Work items and Issues**

NEW

28

Automations, work items, and issues now expose the composer's Agent chip.

— One-line mention, no behavior detail. `2.2.1`

---

46 **Live sidebar filtering across sessions and conversations** NEW

28

Filter sessions, chats, projects, and Claude Code conversations live with the new Filter sessions field at the top of the sidebar.

— One-line UI addition, no further mechanism. `1.11.0`

└──▷ BREAKING ON UPGRADE

! `alta review` now handles file and change review; `alta ask` no longer accepts a `file` argument. Scripts or prompts that passed `file` to `alta ask` now receive `ask.fileMoved` and must be updated to use `alta review request file` instead.

! A custom `default.prompt.md` that replaces the built-in prompt must retain its `Review by the user` section to remain compatible with the new review workflow.

! After upgrading, each existing session restarts its tools once because the tool list has changed; on Anthropic, the first tool loaded by tool search costs one prompt-cache fill.

! A session that loaded tools via tool search cannot be opened by an older version of CodeAlta.

WAS THIS USEFUL? 👍 👎

keploy

Feedback ⓘ v3.6.179 NOTES ↗ RANK

Open-source platform for creating safe, isolated production sandboxes for API, integration, and E2E testing.

> Keploy sharpened its recording and replay fidelity this window: replay now blocks real network traffic by default for fully offline test runs, while recording improved time-zone accuracy, decoded-body capture, and browser compatibility via CA trust and CORS preflight handling.

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

01 Replay blocks real network access by default

BREAKING

64

Keploy's replay mode (`keploy test`) now blocks real outbound network traffic by default in userland, ensuring tests run against mocks only and never reach live dependencies. This is a breaking change: any test that previously relied on live network access during replay will fail unless the relevant traffic is captured as a mock.

— Clear before/after and breaking impact, but no migration flag given v3.6.174

THINNER COVERAGE BELOW

02 Container time zone detection for replay accuracy

IMPROVED

45

Keploy now reads each app container's time zone through Docker in docker and compose modes, so time-based test replay stays accurate when containers run in non-UTC zones.

— Explains mechanism and scope but no config surface to act on v3.6.176

03 Decoded body weighing in record mode

IMPROVED

45

Keploy's `record` command now lets the caller weigh decoded bodies and reserves buffer space for them as each body is decoded, improving capture accuracy for large or streaming payloads.

— Names the command and mechanism but no exact flag or limit v3.6.179

04 Browser recording support: CA trust and CORS preflight

IMPROVED

44

Keploy now registers its CA into the NSS database so Chromium and Firefox automatically trust it during recording sessions without manual certificate installation, and serves a CORS preflight response as a phase of the HTTP call it guards so browser-based apps are captured without CORS errors interrupting the flow.

— Two concrete browser fixes but no commands or config to invoke v3.6.171

└──▷ **BREAKING ON UPGRADE**

! Replay (`keploy test`) now blocks real outbound network calls by default; any test that previously relied on live network access during replay will fail unless the relevant traffic is captured as a mock.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.166.0 NOTES ↗ CODE ↗

RANK

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

> Codex CLI's rust-v0.166.0 release adds thread-resume and network-only diagnostic subcommands, keyboard-driven transcript selection in the TUI, GPT-6.1 Terra model support, and clearer safety-block explanations with more consistent Markdown rendering.

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

01 Resume queued threads with codex queue --resume NEW

70

Resume a thread that was unloaded mid-session after queuing a follow-up message using `codex queue --resume`, so the conversation continues without manual re-opening.

Resume a thread that was unloaded mid-session after queuing a follow-up message, so the conversation continues without manual re-opening.

```
$ codex queue --resume
```

— Runnable command given, but mechanism of unloading not detailed rust-v0.166.0

02 Network-only diagnostics via codex doctor NEW

70

Run focused connectivity and WebSocket diagnostics in isolation, without a full configuration check, using `codex doctor --network-only`.

Diagnose connectivity or WebSocket problems in isolation without running a full configuration check.

```
$ codex doctor --network-only
```

— Runnable command, but scope of diagnostics only briefly described rust-v0.166.0

03 Keyboard transcript selection in the TUI NEW

60

Codex CLI now supports keyboard-driven transcript selection with word, line, and page motions, integrated search, and configurable shortcuts in the TUI.

— Describes motions and search, but no exact keys or config syntax given rust-v0.166.0

THINNER COVERAGE BELOW

04 **GPT-6.1 Terra model support** NEW 55

Codex CLI adds GPT-6.1 Terra to the bundled, Vertex AI, and Amazon Bedrock model catalogs.

— Names the model and three catalogs but no usage detail `rust-v0.166.0`

05 **Consistent Markdown rendering in the TUI** IMPROVED 40

Codex CLI now renders Markdown consistently in async questions, choices, answers, and transcript replies in the TUI.

— Names surfaces affected but gives no mechanism or example `rust-v0.166.0`

WAS THIS USEFUL? 👍 👎

flow-next

Feedback ⓘ flow-next-v9.4.0 NOTES ↗ RANK

flow-next is a workflow plugin that takes a bug, idea or ticket to a verified pull request with specs, cross-model review, live QA, and receipts

> flow-next's biggest addition this window is goal mode, which batches multi-child specs onto a shared `goal/<branch>` integration branch for a single final pull request — a routing change now mandatory and requiring a CI update — alongside plain-word rigor control for goal specs, a new `GOAL_MET` completion signal, and a sharp jump in false-claim detection accuracy for subagent checks.

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

01 Goal mode runs multi-child specs as one PR

BREAKING

91

`flow-next:flow --goal` runs a full goal spec as a coordinated batch, routing all children onto a `goal/<branch>` integration branch and producing one final pull request instead of one per child. As of v9.4.0 this routing became mandatory — every child now lands on `goal/<branch>` instead of the default branch — so repositories whose CI triggers only fire on pull requests into the default branch must add `goal/**` to their CI trigger, or the run stops and asks.

Run a six-feature PRD as a single coordinated goal, landing all children on one integration branch with one final review instead of six separate pull requests.

```
$ flow-next:flow --goal
```

— Names the flag, branch pattern, and exact CI trigger change needed. flow-next-v9.4.0

02 Plain-word rigor control in goal specs

NEW

65

Per-child ceremony can be set inline in the goal spec using plain words — ranging from 'commit each feature straight, one review at the end' to the full gates-per-child default — covering pull request per child, review per child, final review, and QA, without any external configuration or new config keys. Where the spec's words conflict with a repository rule such as branch protection, the stricter rule wins.

— Explains mechanism and conflict resolution but no concrete config key. flow-next-v9.4.0

03 Codebase-grounded verification for subagent claims

IMPROVED

60

Subagent capture, audit, and feature-map checks now use the host codebase's own search to verify claims a conversation makes about the code, raising caught false-claim

detection from roughly 0.6 to 0.96 and above.

— Gives concrete before/after metric but no user-facing control surface. product docs

THINNER COVERAGE BELOW

04 **GOAL_MET signal on goal completion** NEW 40

Goals now emit a `GOAL_MET` signal recorded against the collected evidence when the spec is satisfied.

— Names the signal but gives no detail on how it's consumed. product docs

└──▷ **BREAKING ON UPGRADE**

! Every child in a goal now lands on `goal/<branch>` instead of your default branch; CI triggers scoped only to pull requests into the default branch will stop the run.

WAS THIS USEFUL? 👍 👎

harness-remote

Feedback ⓘ v3.7.0 NOTES ↗ RANK

Native-session control plane for Codex, Claude Code, OpenCode, OMP and PI. Run, resume and hand off coding sessions across your machines.

> Harness Remote v3.7.0 introduces a Plugin Center for managing machine-local plugins and ships Token Harness, its first plugin, for tracking Claude Code and Codex token allowances.

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

01 **Token Harness plugin for allowance tracking** NEW 75

Token Harness is the first plugin for the new Plugin Center, showing Claude Code and Codex five-hour and weekly allowance windows, locally recorded token savings, and model-routing switches directly in plugin settings. Its dashboard includes an Open Token Harness button giving access to RTK, HarnessTrim, and other optimization controls.

Install and enable the Token Harness plugin to monitor Claude Code and Codex token allowances from inside Harness Remote.

> 📍 In the app, go to Settings → Plugins, select the target machine, locate Token Harness, and toggle it to enabled. The dashboard then shows five-hour and weekly allowance windows alongside locally recorded token savings.

— Concrete windows and controls named, with steps but no CLI/API v3.7.0

02 **Plugin Center for machine-local plugins** NEW 70

A new Plugin Center, reached via Settings → Plugins, lets users choose a machine, inspect installed plugins, view diagnostics, and manage supported actions. It supports built-in and user-installed Node or Python plugins with persisted settings and explicit enable/disable controls; user-installed plugins do not run until enabled.

— Names UI path and plugin runtime but no API/config keys v3.7.0

WAS THIS USEFUL? 👍 👎

claude-mem

Feedback ⓘ v15.6.0 NOTES ↗ RANK

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

> claude-mem's two releases focus on UI state persistence and save reliability for the My Work and My Rules views, alongside opt-in OpenAI-compatible batching in the Observer and a shift to explicit summary provider selection.

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

01 **URL-based state persistence for My Work view** NEW 90

The My Work view now encodes page, focus area, and open request drawer in the URL via `?view=`, `?focus=`, and `?request=` query parameters, making browser reload and back/forward navigation work correctly. List view, filter, and sort preferences are now remembered between visits using browser storage, falling back to session-only memory when storage is blocked, and the `?theme=` query parameter continues to control theme alongside the system preference.

— Names every query parameter and the storage fallback behavior. v15.6.0

02 **My Rules save concurrency and tracking** IMPROVED 70

My Rules now enforces one save at a time, disabling controls while a save runs, and Undo is refused once the rule has changed since the Approve/Reject receipt was issued. The worker's rule-save response now returns `savedRuleId` for tracking which rule version was persisted.

— Explains save mechanism and names the returned field. v15.6.0

THINNER COVERAGE BELOW

03 **Rules file included in feed signature** IMPROVED 40

The feed signature now includes `rulesFile`, making the rules file part of feed identity.

— Names the field but gives no further mechanism. v15.6.0

04 **OpenAI-compatible batching in Observer** NEW 35

The Observer now supports opt-in OpenAI-compatible batching for memory processing.

— Names the capability but no flag or config to enable it. v15.5.0

05 **Explicit summary provider selection in Work view** BREAKING 35

Work view now requires explicit summary provider selection instead of inferring it automatically.

— Describes the behavior change but not the new selection UI or options. v15.5.0

06 **Cross-tab broadcast for Load in Claude Code** IMPROVED

25

The Load in Claude Code action now publishes the change to other open tabs via cross-tab broadcast.

— States the behavior with no mechanism or scope detail. v15.6.0

WAS THIS USEFUL? 👍 👎

Red Hat ripwire

Feedback ⓘ v0.6.9 NOTES ↗ RANK

With ripwire, coding agents get a ranked repository map showing call graphs, blast radius and test coverage.

> ripwire's v0.6.9 release adds a `--defs` batch name-resolution verb, expands `--for` to answer error-shaped and graph-style questions with richer output, introduces owner-qualified selectors and new hedged-routing behaviour for queue dispatch, and ships several breaking changes to caching and selector formats.

└──▷ **WHAT SHIPPED · 12 FEATURES** most completely described first ? what's the number?

01 **`--defs` verb for batch name resolution** NEW 93

ripwire adds a new `--defs=NAME[,NAME…]` verb that resolves up to 16 names in one call, returning owner-qualified kind, `file:line`, language, source-or-test flag, caller floor, and a `sel=` spelling that picks each definition alone.

Resolve up to 16 ambiguous names in one call to see exactly which definition each refers to and get a pasteable `sel=` selector for follow-up queries.

```
$ ripwire --defs=parse,render,run
```

— Exact flag, runnable example, and output fields all given v0.6.9

02 **Hedged routing refinements for queue and name-key hops**

IMPROVED 85

Hedged hops (`via="name"` , `via="name-key"` , `via="queue"` , `via="slot"`) are now excluded from `reaches=` , `reachable=` , and `hops=` counts, which hold proven rows only. Python message-queue dispatch now traces a `put` hand-off to `get` readers via `via="queue"` , and posted message classes name their handlers via `<handler_rule>` on `--path` with `via="name-key"` .

Trace a call path through a Python message queue while printing the bodies of each hop, then disable hop-body printing if the output is too large.

```
$ RIPWIRE_NO_HOP_BODY=1 ripwire --path 'post_message'
'_dispatch_message'
```

— Named via= types, counts, and a runnable path example v0.6.9

03 **Body printing for `--path` middle hops and unreached routes**

NEW

83

`--path` now prints the body of its middle hop (up to 4,000 bytes, truncated and noted beyond that), and an unreached `--path` that names a route prints the bodies of bridge rows and readers marked `proven="0"`.

Trace a call path through a Python message queue while printing the bodies of each hop, then disable hop-body printing if the output is too large.

```
$ RIPWIRE_NO_HOP_BODY=1 ripwire --path 'post_message'
'_dispatch_message'
```

— Exact byte limit and flag, demonstrated by a runnable example v0.6.9

---

04 **Error-shaped `--for` questions with richer graph answers**

IMPROVED

80

`--for` now enables error-shaped questions (tasks phrased with `error`, `exception`, `panic`, `raise`, `throw`, `fail`, `crash`, `Traceback`, or an error-type identifier) to carry a `<callers>` chain marked `by="rank"` even when no function is named; its answers now include `<callers>`, `<graph_verb>`, `<ep>`, `<alts>`, and `<bodies>` elements, plus `slots=`, `mem=`, and `ctor_l=` attributes, and a `stem="1"` marker on the map. A `--for` task now also follows a one-line wrapper into the function it calls, names `ctor_calls=` for the top method's class constructor, reports `slots=` for table-initializer bindings, and answers a task spelled as a graph question (`path from A to B`, `callers of X`) with that verb's answer, honouring `--token-budget`.

— Many named fields but no runnable example of the new behaviour v0.6.9

---

05 **Kill-switches for new hop-body, ctor-hop, and map-stem output**

NEW

80

Turn off new hop-body printing, constructor-hop expansion, or map-stem rendering individually with environment switches `RIPWIRE_NO_HOP_BODY=1`, `RIPWIRE_NO_CTOR_HOP=1`, or `RIPWIRE_NO_MAP_STEM=1`.

Trace a call path through a Python message queue while printing the bodies of each hop, then disable hop-body printing if the output is too large.

```
$ RIPWIRE_NO_HOP_BODY=1 ripwire --path 'post_message'
'_dispatch_message'
```

— Three exact environment variables with a runnable example v0.6.9

---

06 **Cache and snapshot format bump forces re-index**

BREAKING

70

Every ingest cache and quality snapshot written by an earlier build is refused and re-indexed on first run (parser version 259, cache version 46, quality-snapshot scheme 24).

— Exact version numbers but re-index happens automatically v0.6.9

07 **Owner-qualified selectors on member and signature rows** NEW

68

Member rows on `--callers`, `--callees`, `--impact`, `--path` and signature rows now print their owner (e.g. `Pattern.match`, `Builder::Build` in C/C++/Rust, `Task#execute` in Ruby) as a pasteable selector.

— Named flags and example selectors but no runnable command v0.6.9

08 **`--dead-code` test-only caller detection** NEW

65

`--dead-code` adds a new `<to>` row group after its `<d>` rows for library functions whose every indexed caller is test code; these are excluded from `count=`.

— Named flag and output field but no example invocation v0.6.9

09 **`--grep` regex-interpretation reporting** IMPROVED

65

`--grep` now reports `pattern_meta=` when a literal pattern is read as a regex, instead of returning `hits="0"` silently.

— Named flag and field but no example v0.6.9

10 **`file:LINE` ambiguity now refuses with candidate list**

65

BREAKING

A `file:LINE` selector where two definitions start on the same line now refuses with a list of candidates on stderr instead of silently binding the innermost definition.

— Named selector format and new stderr behaviour described v0.6.9

11 **`--affected` hedged group for test class construction** NEW

60

`--affected` adds a separate hedged group marked `via="class-new"` after proven rows for tests that construct a reached class.

— Named flag and marker but no example v0.6.9

THINNER COVERAGE BELOW

12 **Signature row format folds owner into `n=`** BREAKING

55

On a signature `<d>` row, the owner is now folded into `n=` and `sc=` is dropped.

— Exact field change named but no migration guidance given v0.6.9

└──▷ BREAKING ON UPGRADE

! Every ingest cache and quality snapshot written by an earlier build is refused and re-indexed on first run (parser version 259, cache version 46, quality-snapshot scheme 24).

! On a signature `<d>` row, the owner is now folded into `n=` and `sc=` is dropped.

! A `file:LINE` selector where two definitions start on the same line now refuses with a list of candidates on stderr instead of silently binding the innermost definition.

WAS THIS USEFUL?  

Charm Crush

Feedback ⓘ v0.99.0 NOTES ↗ CODE ↗ RANK

Crush is a terminal-based AI coding assistant that integrates LLMs, LSPs, and MCP servers into your development workflow.

> Crush v0.99.0 overhauls how providers and models are configured with the Catwalk v3 type system, expands lifecycle hooks beyond tool events, and gives host applications a richer client/server API including permission piping and server reload support.

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

01 Catwalk v3 provider type system

BREAKING

90

`crush provider add` now uses `--type completions`, `--type responses`, or `--type messages` to describe the API format rather than the vendor. This replaces the old provider `type` values `openai`, `openai-compat`, and `anthropic`; existing configs are migrated automatically on load, so no manual edits are required.

Register a DeepSeek provider and a model with reasoning and attachment limits using the new Catwalk v3 type system

```
$ crush provider add deepseek \
  --type completions \
  --base-url "https://api.deepseek.com/v1" \
  --api-key "$DEEPSEEK_API_KEY"

crush model add deepseek/deepseek-chat --thinking toggleable --
max-attachments 5
```

— Full command example plus exact flag values and migration note. v0.99.0

02 Per-model reasoning and attachment limits

BREAKING

84

`crush model add` now supports `--thinking always|never|toggleable` to control per-model reasoning and `--max-attachments N` to set attachment limits; this replaces the old `--can-reason` flag.

Register a DeepSeek provider and a model with reasoning and attachment limits using the new Catwalk v3 type system

```
$ crush provider add deepseek \
  --type completions \
  --base-url "https://api.deepseek.com/v1" \
  --api-key "$DEEPSEEK_API_KEY"

crush model add deepseek/deepseek-chat --thinking toggleable --
max-attachments 5
```

— Names exact flags, values and the flag they replace. v0.99.0

03 **Pipe permission prompts to host application** NEW 75

Set `CRUSH_PERMISSIONS_PIPED=1` to pipe permission prompts to the host application instead of showing them interactively, useful when Crush is embedded in an editor or service.

Let a host application handle permission prompts so users are not asked twice when Crush is embedded in an editor or service

```
$ CRUSH_PERMISSIONS_PIPED=1 crush
```

— Exact env var with a runnable command example. v0.99.0

04 **Lifecycle hooks for session, turn and permission events**

IMPROVED 65

Hooks now cover session, turn, permission, and compaction lifecycle events in addition to tool events, including `PostToolUse` and `PostToolUseFailure`. Hook configurations are validated against a list of known events, so typos are caught before a run starts.

— Names event types but no runnable example given. v0.99.0

05 **Server reload and connection alias** IMPROVED 60

Send `SIGHUP` to reload Crush's config without stopping the server, and use `http://` as an alias for `tcp://` when starting the Crush server.

— Names exact signal and URL scheme alias, no example run. v0.99.0

THINNER COVERAGE BELOW

06 **Richer client/server API for host apps** IMPROVED 55

In client/server mode, the host application can now supply context (such as open files) that appears in completions, theme the TUI, append text to the next prompt, and see which session other clients are viewing.

— Describes capabilities without naming the concrete API surface. v0.99.0

└──▷ **BREAKING ON UPGRADE**

! Provider `type` values `openai`, `openai-compat`, and `anthropic` are replaced by `completions`, `responses`, and `messages`; existing configs are migrated automatically on load, so no manual edits are required.

! The `--can-reason` flag on `crush model add` is replaced by `--thinking always|never|toggleable`.

WAS THIS USEFUL?  

Alibaba Qwen Code

Feedback ⓘ sdk-typescript-v0.1.26

NOTES ↗ RANK

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

> Qwen Code's biggest change this window is a new `settings.env` field for injecting environment variables into agent sessions and multi-modal (image, PDF, audio) input support, alongside several breaking changes — a `disable*` → `enable*` settings rename, removal of the `read_many_files` tool, and removal of Smart Edit and `ClearcutLogger` — plus incremental improvements to Generated UI, the web shell, managed-agent worktrees, PR verification, Autofix, and extension installation.

└──▷ **WHAT SHIPPED · 12 FEATURES** most completely described first ? what's the number?

01 Custom environment variables via settings.env NEW

78

A new `settings.env` field in the settings file lets users inject custom environment variables — such as API keys or proxy settings — into every agent session without modifying the shell environment.

Inject API keys or proxy settings into every agent session without modifying the shell environment.

```json
{
  "env": {
    "HTTPS_PROXY": "http://proxy.corp.example.com:8080",
    "MY_API_KEY": "sk-..."
  }
}
```

— Named config key with a runnable JSON example. sdk-typescript-v0.1.26

THINNER COVERAGE BELOW

02 Web shell timing, session list, and workspace selection

IMPROVED

59

The web shell now shows model timing inline in the Token Usage display, moves no-workspace sessions beside workspaces in the session list, and lets embedding hosts pick workspace directories for web-shell sessions.

— Names the UI field changed but no integration API given. desktop-v0.26.4

03 **Independent worktrees and phased upgrade for managed agent**

IMPROVED

55

The managed agent now creates independent worktrees for main sessions, isolating each session's file tree, and adds a phased upgrade command to the reference distribution.

— Names mechanism but not the upgrade command itself. v0.26.4 desktop-v0.26.4

04 **Install extensions from unscoped npm packages** IMPROVED

53

Extensions can now be installed and updated from unscoped npm packages, removing the requirement for scoped package names.

— Names the change but gives no install command. v0.26.4

05 **Autofix round budget and worktree file-system watches**

IMPROVED

52

Autofix improvements now bound autofix takeover with a round budget in the CI pipeline, and add root file-system watches at the session worktree under the daemon in Autofix PR.

— Names mechanisms but no exposed configuration values. desktop-v0.26.4

06 **Interactive views and theming in Generated UI** IMPROVED

48

Generated UI now presents interactive views when users want to adjust values, automatically follows the host application theme, and localizes and reshapes card frames for consistent presentation.

— Describes UI behavior but no config or API surface. desktop-v0.26.4

07 **Version stamping and re-verify trigger in PR verification**

IMPROVED

48

PR verification now stamps model, CLI, and skill version on published reports, and adds re-verify after update as a follow-up trigger.

— Names stamped fields but no command or endpoint shown. desktop-v0.26.4

08 **Smart Edit tool and ClearcutLogger removed** BREAKING

45

The Smart Edit tool and `ClearcutLogger` are removed; workflows depending on either will break.

— Names removed components but no replacement or migration path. sdk-typescript-v0.1.26

09 **Configurable quoted replies in DWS channel** NEW 38

The DWS channel now supports configuring quoted replies by conversation type.

— Names the feature but no config key or command. desktop-v0.26.4

10 **Default cursor and macOS Picture-in-Picture for computer use** NEW 38

Computer use sessions now enable a default cursor and live macOS Picture-in-Picture.

— Names the feature but no setup or trigger steps. v0.26.4

11 **Raised workspace skill install limits in CLI** IMPROVED 35

The CLI now raises the number of workspace skills that can be installed.

— States the change without a new limit number. v0.26.4 desktop-v0.26.4

12 **Extended PATH for macOS desktop daemon** IMPROVED 27

The desktop daemon spawned on macOS now has an extended PATH.

— Thin description with no specifics on paths added. desktop-v0.26.4

└──▷ BREAKING ON UPGRADE

! Settings keys named `disable*` are renamed to `enable*` (e.g. `disableFoo` becomes `enableFoo`) — any config files or scripts referencing the old negative-named keys will stop working.

! The `read_many_files` tool is removed; its functionality is now a `readManyFiles` utility for user @`-commands` only — automations or skills that called `read_many_files` as a tool will break.

! The Smart Edit tool and `ClearcutLogger` are removed — workflows depending on either will break.

WAS THIS USEFUL? 👍 👎

termide

Feedback ⓘ 0.47.0 NOTES ↗ RANK

All-in-one terminal workspace for desktops and servers: editor with LSP, file manager with SFTP/FTP, terminal, git, database viewer and a coding agent in one zero-config static Rust binary.

> Termide's six releases add a full DAP debugging panel with per-debugger configuration and zero-config Rust support, a plugin system, and a reworked agent-delegation stack (external-agent delegates, pause/survival, and rate-limit parking), alongside SFTP host-key verification and `ssh_config` support, inline command placeholders, and image rendering in the agent transcript.

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

01 DAP debugging with per-debugger configuration NEW

98

Set breakpoints with `Ctrl+F8` or by clicking a line number in the editor gutter; they persist per file across restarts. Start a command-line debugger with `F5`, naming it in `commands.toml` with `debugger = "lldb"`, `"gdb"`, `"dlv"` or `"pdb"` and an optional `build` key (Python and Go need no entry at all); add `console = "panel"` to keep debugged-program output in the debug panel. Control the debugger from the agent panel with the `debug` tool — set breakpoints, start a session, step, select frames, and evaluate expressions, each response returning the current stop location, call stack, variables, and last output — and manage sessions via `Start debugging` and `Remove all breakpoints` in the command palette or editor `[≡]` menu. Per-debugger configuration was then added via `[debug]` and `[debug.debuggers.<name>]` in `config.toml` (`command`, `args`, `native` order for Rust, `console` adapter default), `debugger_args` in `commands.toml` to pass flags directly to the adapter, a guided `Set up debugging` form that saves a launch to `.termide/commands.toml`, and zero-config Rust debugging from `F5` on a Cargo project (building with `cargo build -p <package> --bin <name>` and launching under `lldb-dap`, GDB's adapter, or `lldb`). The `debug start` tool now also accepts commands that take `params` by name, and stepping, pausing, stopping, breakpoints, and frames no longer prompt outside plan mode.

Add verbose DAP logging to a specific launch command so the flags reach the adapter without touching the global debugger config.

toml

```toml
[[commands]]
name = "Debug server"
program = "./server"
debugger_args = ["--log", "--log-output=dap"]
```

— Full mechanism and every named config key and shortcut given 0.47.0 0.44.0 0.43.0

02 `on_limit` rate-limit handling for runs and delegates

IMPROVED

83

Setting `on_limit = "ask_else_wait"` shows a failure card and automatically retries once the reset time passes, so unattended `/goal` or `/loop` runs are not left waiting. This was extended so delegate tasks that hit a quota, rate, or transient limit park instead of failing, shown with a `⏸` indicator naming the delegate and resume time and waking automatically at reset, on request, or when the panel's connection changes; `on_limit` can prompt to retry or give up, and after six retries the limit is surfaced to the delegating agent naming the connection and reset time.

Let an unattended `/loop` survive API rate-limit errors by retrying automatically once the reset window passes, without leaving the run stalled at a card.

toml

```toml
# In .termide/config.toml:
[ai]
on_limit = "ask_else_wait"
```

— Exact config key, values and retry count given 0.46.0 0.42.0

03 Automatic reasoning parameter negotiation

BREAKING

82

A `send_reasoning` connection setting (`auto` , `on` , `off`) was added to pass reasoning back to OpenAI-compatible local models, using `reasoning_content` and `preserve_thinking` for models such as Qwen 3. This was replaced by automatic per-server negotiation: OpenAI, OpenRouter, and the Gemini API now receive `reasoning_effort` , while other servers receive `chat_template_kwargs.enable_thinking` , `reasoning_content` , `reasoning` , and `preserve_thinking` with no manual settings needed; the `reasoning_param` and `send_reasoning` connection settings (Reasoning parameter , Send reasoning back) are removed and silently ignored in existing configs.

Enable reasoning passback for a local Qwen 3 endpoint so earlier thoughts are preserved across turns and answer quality does not degrade.

toml

```toml
# In the connection's settings (ai section):
[ai]
send_reasoning = "on"
```

— Before/after settings and parameter names fully specified 0.43.0 0.42.0

04 **Plugin system for commands and agent tools** NEW 80

Termide now supports plugins: drop a program into `plugins/<name>/` beside the global `config.toml` with a `plugin.toml` naming it, running per-project over JSON-RPC on stdin/stdout to add commands, status-bar items, editor line marks, agent tools, and more (see `doc/en/plugins.md` and `examples/plugins/`). Each plugin's mode — `on`, `opt-in`, or `off` — is controlled in the global `config.toml`, with per-project overrides in `.termide/config.toml`.

— Protocol, file layout and config keys all named `0.45.0`

---

05 **`--detached` / `--no-attach` instance flags** BREAKING 80

Start and immediately attach to a detached instance with `termide --detached` (short form `-d`); use `--detached --no-attach` to start without attaching and print the instance id for scripts. This changes prior behaviour: scripts that ran `--detached` to start only must add `--no-attach` to preserve the old start-only behaviour.

Start a persistent workspace you can disconnect from and reconnect to later, without losing shells or agent work.

```
$ termide --detached
```

Start a detached instance from a script or a launch without a terminal, printing its id for later `--attach`.

```
$ termide --detached --no-attach
```

— Exact flags with runnable commands and migration note `0.47.0`

---

06 **Delegate run-location setting added then removed** BREAKING 78

A `subagent_run` setting (`hidden`, `existing_panel`, `panel`, or `ask`) in the `[ai]` settings or a delegate's `AGENT.md`, plus a `background` flag on `task` to run delegated tasks in the background and return control immediately, was added and then removed in the same window: the `[ai] subagent_run` key in `config.toml` is now silently ignored, the `subagent_run` key in `AGENT.md` is reported as unread, the Delegated tasks run settings option is gone, and all delegates now unconditionally run out of sight.

Route all delegated tasks into visible agent panels so you can watch their progress and answer permission prompts, rather than having them run hidden.

```toml
# In .termide/config.toml or the AI settings tab:
[ai]
subagent_run = "panel"
```

— Exact keys and removal behaviour named with example `0.45.0` `0.42.0`

07 **Inline placeholders in command definitions** IMPROVED

78

Use `{file}`, `{line}`, `{col}`, `{files}`, `{dir}`, and `{project}` placeholders directly in a command's `command` or `build` field so values are substituted before the command runs, replacing the previous requirement to reference `TERMIDE_PARAM_*` environment variables that PowerShell and cmd cannot read.

Run a Perforce edit command on whichever file the editor cursor is in, without typing or passing a path manually.

```toml
command = "p4 edit {file}"
```

Open a compiler at the exact file and line where the editor cursor sits, passing both as inline placeholders.

```toml
command = "gcc -c {file} && echo 'line {line}'"
```

— Named placeholders and before/after with runnable examples `0.46.0`

08 **SFTP host-key verification** NEW

78

SFTP now checks server keys against `known_hosts` before sending credentials, prompting to accept unknown keys (showing type and SHA256 fingerprint) with Accept and remember, Accept once, or Cancel; changed or revoked keys are refused with the offending `known_hosts` line and the `ssh-keygen -R` command to fix it.

— Full verification flow and remediation command named `0.47.0`

09 **Agent job control (`jobs`, `stop`)** NEW

75

Two new agent tools control background work: `jobs` lists running background tasks, parked tasks, and backgrounded calls each with an id and elapsed time, while `stop` cancels one by id without prompting — visible in the agent transcript as `& Listing background work` and `& Stopping <id>`.

— Tool names and transcript output named, no CLI example `0.47.0`

10 **Delegate pause, survival, and external delegation** NEW 73

Pausing a panel now holds its delegates before their next request, shown as `|| Paused · 2 waiting` in the state strip and resumed with `/continue` or `[▶]`. Delegated tasks now survive termide closing and are carried on by the originating panel, appearing as `⏸ &` in session lists (`F6`) and as rows above the input. External agents with a `command` field can now be used as delegates via `task`, running out of sight with their own program, tools, and permission cards routed to the delegating panel; an agent can also delegate to a fresh copy of itself via `task` for background or parallel work.

— Behaviour and UI markers named, no runnable command `0.45.0`

---

11 **Image reading and transcript rendering** NEW 73

The `read` tool now returns PNG, JPEG, GIF, and WebP files as images scaled to 1568 px instead of refusing them as binary, so the agent can inspect screenshots and rendered output. Those images render directly in the agent transcript via Kitty, Sixel, iTerm2, or half-block fallback: a picture returned by `read` appears under its call, and a pasted image appears inside the message, centred and capped at half the panel's height or width.

— Formats, scaling and renderers named, behaviour is automatic `0.47.0` `0.42.0`

---

12 **Private skills via front matter** NEW 73

Adding `disable-model-invocation: true` front matter keeps a skill private to the user and out of the model's toolset while still runnable via `/name`. Skills are also now read from `~/.agents/skills/` as the lowest skill level, picking up skills installed globally with `npx skills add -g` or by Codex and other agents.

— Exact front matter key, path and install command named `0.42.0`

---

13 **Inter-panel agent messaging** NEW 71

New `agents` and `send_message` tools let agents in different panels list peer panels and send `info`, `question`, or `reply` messages; an agent now refuses an edit or write to a file changed by another panel, telling it to re-read first to prevent conflicting concurrent edits. These exchanges now display as steps under `&` in the activity plate, with labels such as `& Receiving reply from default (panel 3) · …`, `& Asking panel 3 · …`, and `& Listing agent panels`, replacing the previous `@` glyph.

— Named tools and UI labels, but no direct command `0.44.0` `0.42.0`

---

14 **SFTP `ssh_config` support** NEW 70

SFTP now reads `ssh_config` as OpenSSH does — user and system files, `Host` aliases, `HostName`, `Port`, `User`, `ProxyJump`, `UserKnownHostsFile`, `GlobalKnownHostsFile`, `HostKeyAlias`, `HashKnownHosts`,

`StrictHostKeyChecking yes`, `Include`, `!` patterns, and `%h` / `%p` / `%r` token expansion.

— Every supported directive named, nothing to run directly `0.47.0`

15 **Directory-tree destination picker** NEW — 70

Pick copy, move, pack, unpack, and Save As destinations from a directory tree dropdown — `Tab`, click, or arrow to open; `↓` to enter; `Enter` to confirm; `Esc` to cancel — with fuzzy filtering, recent destinations, and inline directory create/rename/delete.

— Exact keys and capabilities named, clear navigation path `0.47.0`

16 **Syntax highlighting performance and file-size limit** IMPROVED — 68

Syntax highlighting now re-parses only the edited region and highlights only what is on screen, cutting highlight time from ~1 s to ~1 ms per keystroke in large files; the file-size limit for highlighting rises from 1 MiB to 2 MiB.

— Concrete before/after numbers, nothing for user to configure `0.47.0`

17 **`cwd = "project"` command setting** NEW — 67

Set `cwd = "project"` in `commands.toml` to run any command from the project root regardless of which panel is focused, or tick the `Run from the project folder` box in the command's create/edit form.

Pin a deploy script to the project root so it always resolves paths correctly, no matter which subdirectory panel is focused.

```toml
[commands.deploy]
cmd = "./scripts/deploy.sh"
cwd = "project"
```

— Exact config key and example given `0.45.0`

THINNER COVERAGE BELOW

18 **Mermaid panning and output folding** IMPROVED — 57

Mermaid diagrams wider than the agent panel now pan horizontally via the scroll wheel, `Shift` +wheel, click/drag on the scrollbar, or `Shift+→` / `Shift+←`. Long Mermaid diagrams (taller than half the agent panel) and `write` output for HTML, Markdown, or Mermaid files now fold to a one-row card; click or `o` opens the file, `Alt` +click opens it outside Termide.

— Interaction keys named, impact is cosmetic/UX only `0.47.0`

19 **Session history search** NEW — 55

Search session history from the agent banner with `Ctrl+F`, which turns the prompt box into a fuzzy session search with match highlighting and arrow-key navigation.

— Exact shortcut given, behaviour described briefly `0.42.0`

20 **Headless web search** IMPROVED 50

`web_search` now works without a browser by fetching the results page over plain HTTP, making it available in Termux and headless servers wherever a search engine is configured; DuckDuckGo (the default) works this way.

— Mechanism and default engine named, no config shown `0.45.0`

21 **MCP read-only tools skip prompts** IMPROVED 50

MCP tools marked read-only (`readOnlyHint`) now run in plan mode without a prompt, and in `edit` mode without asking.

— Named field and modes, but thin on mechanism `0.42.0`

22 **Repository dropdown tree rendering** IMPROVED 45

Repository dropdowns now render the project's repository tree with fold/expand controls (`Enter`, `←` / `→`, or click) so multiple same-named repos across hosts and groups are distinguishable.

— UI behaviour and keys named, limited scope described `0.45.0`

23 **Context trimming on compaction failure** IMPROVED 42

The context is now cut to fit within the model window without a model call when compaction fails, prioritising removal of reasoning, long output, and old exploration over recent turns.

— Behaviour described, no surface for user to act on `0.42.0`

24 **`prefill_progress` connection setting removed** BREAKING 40

The `prefill_progress` connection setting (shown in the UI as `Prefill progress (llama.cpp)`) is removed; existing configs that name it still load and the key is silently ignored.

— Names the key but gives no replacement behaviour `0.43.0`

└──▷ **ALSO FROM THESE RELEASES**

.
.
.
.
.
.

Skip automatic project restore for a one-off launch into a specific directory without reopening background projects.

```
$ termide --no-restore /path/to/project
```

└──▷ **BREAKING ON UPGRADE**

! `termide --detached` now starts the instance AND attaches immediately; scripts that ran `--detached` to start only must add `--no-attach` to preserve the old behaviour.

! The `[ai] subagent_run` key in `config.toml` is now ignored; set it and termide silently discards it. The `subagent_run` key in `AGENT.md` is reported as a key termide does not read. All delegates now unconditionally run out of sight.

! The `reasoning_param` and `send_reasoning` connection settings (shown in the UI as `Reasoning parameter` and `Send reasoning back`) are removed; existing configs that name them still load and the keys are silently ignored.

! The `prefill_progress` connection setting (shown as `Prefill progress (llama.cpp)`) is removed; existing configs that name it still load and the key is silently ignored.

WAS THIS USEFUL? 👍 👎

wrongstack

Feedback ⓘ v1.3.0 NOTES ↗ RANK

An AI coding agent that reads your code, edits files, runs commands, and reasons through bugs — across a terminal REPL, a full-screen TUI, and a browser UI, while you keep your hand on every permission.

> WrongStack ships a new Desktop v2 source workspace, reproducible agentic benchmarks with a lean tool profile, and finer-grained instruction and prompt-journal controls, alongside standalone binaries for all major platforms.

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

01 **Desktop v2 source workspace** NEW 75

WrongStack adds a Desktop v2 source workspace with a durable project, roster, mailbox, automation, session model-plan and skill drafts, virtualized file navigation with bulk file operations, file and line-range references, reviewed file writes, and Git hunk/conflict review.

— Names many concrete capabilities but no commands or config keys given v1.3.0

02 **Reproducible agentic benchmarks with lean tool profile** NEW 69

WrongStack adds reproducible agentic benchmarks with model and prompt comparisons, Polyglot subsets, resume/rerun, sandbox cleanup, and a lean profile restricted to six file/shell tools without lazy schema expansion. Benchmark timeout and cancellation now terminate the complete POSIX process group, including descendants that keep output pipes open.

— Describes mechanism and scope but no runnable command shown v1.3.0

03 **Standalone binaries for all major platforms** NEW 60

Standalone binaries are now available for Windows x64/ARM64, macOS x64/ARM64, and Linux x64/ARM64 with glibc and musl variants.

— Specific platform list, clear download target, no further detail v1.3.0

THINNER COVERAGE BELOW

04 **Global and project AGENTS.md instruction scoping** NEW 58

WrongStack now supports separate `global` and `project` `AGENTS.md` switches to control instruction scope independently per interface.

— Names exact config switches but no file path or example given v1.3.0

05 **Persistent media blocks across tools and providers** IMPROVED

57

Image attachments and tool-produced screenshots now retain their media blocks across tools, MCP, providers, and the WebUI, with model capability routing using the configured provider and catalog identities.

— Explains behaviour and scope but no config surface to act on v1.3.0

---

06 **Quick system prompt and Companion controls** NEW

35

WrongStack adds a Quick system prompt control and Companion controls across interfaces for focused instruction management.

— Thin description, no UI path or config name given v1.3.0

---

07 **Prompt journal filtering, export, and paging** IMPROVED

35

Prompt journals now support month filtering, export, and bounded paging.

— States what changed but no export format or command named v1.3.0

WAS THIS USEFUL? 👍 👎

octomind

Feedback ⓘ 0.66.3 NOTES ↗ RANK

Open-source AI coding agent and agent runtime: one binary, any model, MCP-native. Runs in terminal, CI, or as a daemon.

> Across twelve releases, octomind expanded its WebSocket/`octomind server` surface with resumable sessions, tap-event streaming, and session command dispatch, added interactive MCP elicitation and tool annotations, shipped Windows installation support and checksum-verified installers, and rounded out learning, evolution, and config-validation behavior.

└──▷ **WHAT SHIPPED · 17 FEATURES** most completely described first ? what's the number?

01 Installer Enhancements NEW

80

Releases now publish SHA-256 checksums verified by `install.sh`, and octomind can now be installed on Windows via `install.ps1` through PowerShell (`irm https://raw.githubusercontent.com/muvon/octomind/master/install.ps1 | iex`) or the Scoop package manager.

Install octomind on a Windows machine without WSL, directly from PowerShell.

```
$ irm https://raw.githubusercontent.com/muvon/octomind/master/install.ps1 | iex
```

— Names the exact script and checksum algorithm used in installs. Gives the exact runnable install command for Windows.

0.65.1 0.58.8

02 WebSocket tap-event streaming becomes fallible and race-safe

65

BREAKING

Tap run events can now be streamed over WebSocket using the `octomind server` surface, enabling IDE plugins and dashboards to consume live agent run events. This makes tap run events fallible and race-safe, so code that treated run-event delivery as infallible will need to handle errors.

— Names the server surface and consumers but not an event schema. 0.59.0

THINNER COVERAGE BELOW

03 WebSocket Session Enhancements NEW

57

WebSocket sessions now expose session commands to remote clients connected to `octomind server`, summarized run logs, session control and request-linked costs,

and support resumption so interrupted sessions can reconnect without losing state.

— Groups three thin WebSocket additions without a runnable example. Explains the reconnect mechanism but shows no resumption command.

0.58.6 0.61.0 0.65.0 0.60.0

04 **`--show` now displays configured root model** BREAKING

55

`--show` now displays the configured root model instead of the previous output.

— Names the exact flag and new output, but not the prior behavior. 0.59.0

05 **CLI diagnostics report `LOCAL_API_KEY` status** NEW

55

CLI diagnostics now report the status of the optional `LOCAL_API_KEY` environment variable.

— Names the exact environment variable but not the diagnostic command. 0.58.6

06 **Concurrent execution of read-only tools** IMPROVED 50

octomind now schedules read-only tools concurrently instead of serially, reducing latency when an agent issues multiple safe tool calls in a single turn.

— Explains the serial-to-concurrent change but gives no config knob. 0.66.2

07 **MCP Session and Tool Enhancements** NEW 50

MCP tools gain annotations for richer tool-definition metadata, MCP sessions now expose context epochs, and octomind supports interactive MCP elicitation across client protocols so MCP servers can prompt users for input mid-session.

— Names the capability but not the annotation schema or usage. Explains cross-protocol behavior but no enabling flag is given. Bare mention with no explanation of what a context epoch does.

0.66.0 0.63.0 0.58.6

08 **Hashed identifiers for rendered file lines** IMPROVED

48

Rendered file lines now use hashed identifiers instead of plain line numbers, making file references stable across edits.

— Clear before/after but no example of referencing a hashed id. 0.66.0

09 **Tab-completion for `untap` command** IMPROVED 45

Tab-completion for `untap` now covers tap names, making it easier to remove a previously added registry tap.

— Names the exact command though it needs no further configuration. 0.65.0

10 **Configurable commit attribution** NEW 44

Commit attribution is now configurable via config, and attribution is required when commits are unspecified.

— Names the config mechanism generically without the exact key. 0.61.0

11 **Config validation reports unknown keys** IMPROVED 40

Config validation now reports unknown and silently ignored keys, surfacing misconfiguration that was previously hidden.

— Describes the check but not which config file or command triggers it. 0.59.0

12 **Command aliases shown in help output** IMPROVED 40

The `help` command now includes command aliases in its output.

— Trivial change but names the exact command to try. 0.59.0

13 **Learning gains live-skill recall and multi-pass synthesis** NEW 40

Learning now recalls live skills using persisted experiences, shows in synthesis which evolved artifacts were live, and supports multi-pass learning synthesis.

— Names three learning behaviors but no command or config to trigger them. 0.59.0

14 **Persistent standing session rules** NEW 40

Standing session rules now persist across sessions, so rules defined in one session carry forward into subsequent ones.

— Explains persistence behavior but not where rules are defined or stored. 0.58.5

15 **Provider retries surfaced across clients** IMPROVED 38

Provider retries are now surfaced consistently across all clients, making retry behavior visible regardless of how octomind is invoked.

— States the change but gives no retry count, backoff, or flag. 0.60.0

16 **Supervisor handles unsettled evidence advisories** NEW 25

octomind's supervisor now handles unsettled evidence advisories, expanding the set of advisory states it can process.

— No mechanism or usage detail given beyond naming the advisory state. 0.66.3

17 **Evolution proposal composition and guidance** IMPROVED 23

Evolution now composes compatible evolution proposals and delivers generated evolution guidance.

— Vague prose without naming how proposals are composed or surfaced. `0.59.0`

└──▷ BREAKING ON UPGRADE

! Tap run events are now fallible and race-safe; code that treated run-event delivery as infallible will need to handle errors.

! `--show` now displays the configured root model instead of the previous output.

WAS THIS USEFUL? 👍 👎

cezar

Feedback ⓘ v0.17.0 NOTES ↗ RANK

Open-source orchestrator and ADE for running Claude Code, Codex, OpenCode, Pi and other AI coding agents in parallel — locally or 24/7 on your own server.

> cezar v0.17.0 adds project webhook notifications, self-hosted agent account management, flexible task scheduling, global/organisation-level prompt and skill settings, trunk-based development, and a size-reducing change to the runs API.

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

01 **Run API payload size reduction** BREAKING 90

`GET /api/v1/runs` no longer embeds each run's `workflowDef` (approximately 1.3 MB per run) by default; pass `?full=1` to the endpoint to restore the previous behaviour and include `workflowDef` in the response. `GET /runs/:id` is unchanged.

Retrieve full run definitions from the API when you need workflow details that are now omitted by default.

```
$ curl 'http://localhost:4321/api/v1/runs?full=1'
```

— Exact endpoint, query param, size figure, and scope named v0.17.0

02 **Project webhook notifications** NEW 85

Enable project-scoped webhook notifications for Slack, Discord, or a custom receiver with the `CEZ_WEBHOOKS=1` environment variable; events fire when a task needs attention, reaches review, completes, fails, or is cancelled.

Receive a signed JSON POST to your own endpoint whenever a task completes or fails, without polling.

```
$ CEZ_WEBHOOKS=1 npx cezar-run
```

— Names exact env var and all trigger events v0.17.0

03 **Flexible task start scheduling** NEW 70

Schedule a task's start from the split Send button (Now / Pick date & time / After usage limit resets), or immediately run a queued task over the `maxParallel` cap with the new Start now action.

— UI controls and config cap named, no command shown v0.17.0

04 **Self-hosted agent account management** NEW 60

Enable agent account management on a self-hosted cockpit with the `CEZ_REMOTE_AGENT_ACCOUNTS=1` environment variable.

Let agents manage their own accounts on a self-hosted cockpit without manual provisioning.

```
$ CEZ_REMOTE_AGENT_ACCOUNTS=1 npx cezar-run
```

— Single named env var, no further mechanism given v0.17.0

THINNER COVERAGE BELOW

05 **Global instructions for task prompts** NEW 50

Set global instructions in General settings that are merged into every task's system prompt ahead of the project's own instructions.

— Mechanism described but no config key named v0.17.0

06 **Organisation-wide skills repositories** NEW 50

Add organisation skills repositories in Global settings to load them as team skills across every project, grouped by repository.

— UI path given, no API or config surface named v0.17.0

07 **Mobile swipe-to-run actions** NEW 45

Swipe a GitHub issue or PR row on mobile to start a run with a workflow or skill in one tap.

— Gesture and target named, no further mechanism given v0.17.0

08 **Trunk-based development workflow per project** NEW 40

Use a trunk-based development model per project so tasks work in the checkout and you commit to the base branch after review.

— Behaviour described, no flag or config named v0.17.0

09 **Answer agent question cards** NEW 30

Answer an agent's question card with your own text, or tell the agent you need more context.

— Thin UI description with no mechanism or location v0.17.0

10 **Mermaid diagram previews in conversations** IMPROVED 25

Mermaid diagrams now render as previews inside conversations.

— Bare one-line statement with no further detail v0.17.0

└──▷ **BREAKING ON UPGRADE**

! `GET /api/v1/runs` no longer embeds each run's `workflowDef` (approximately 1.3 MB per run); pass `?full=1` to restore the previous behaviour. `GET /runs/:id` is unchanged.

WAS THIS USEFUL? 👍 👎

zot

Feedback ⓘ v0.4.56 NOTES ↗ RANK

zot manages coding agent workflows from the command line.

> zot adds a Runcrate provider to its built-in AI backend roster.

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

01 **Runcrate AI provider support** NEW 40

zot adds a Runcrate provider, giving it a new built-in AI backend alongside existing providers like Anthropic, OpenAI, and Gemini.

— Names the provider but no config or usage details given v0.4.56

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ Visualize This (2026-10-29)

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 can now render charts, timelines, diagrams, and interactive mockups directly inline within conversation threads.

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

01 **In-thread visualization rendering** NEW 42

Amp now renders charts, timelines, diagrams, and interactive mockups — including clickable prototypes — directly inside a conversation thread in response to natural-language requests.

— Describes what it does but no mechanism or UI path detail. Visualize This (2026-10-29)

WAS THIS USEFUL? 👍 👎

SST OpenCode

Feedback ⓘ v2.2.1 NOTES ↗ RANK

OpenCode is an open-source AI coding agent for terminal-based development tasks.

> OpenCode's latest releases add a goal-tracking session command, experimental image generation that now auto-activates for capable providers, a redesigned model picker, and a unified comment editor for annotations.

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

01 **Goal-tracking sessions via /goal command** NEW 85

Run `/goal <objective>` in the TUI or Desktop to keep a session running until a reviewer confirms the objective is met; control it with `/goal pause`, `/goal resume`, or `/goal clear`.

Keep an agent working toward a long-running task in the TUI until a reviewer confirms it is done.

```
$ /goal implement the OAuth2 login flow and add tests
```

— Names exact command and subcommands, runnable as-is v2.2.1

THINNER COVERAGE BELOW

02 **Redesigned model picker** IMPROVED 40

Search favorites, recent models, providers, and variants from the redesigned model picker, now the default in Desktop.

— Describes UI change but no specific commands or settings v2.2.1

03 **Unified comment editor for quotes and annotations** IMPROVED 35

Quotes and annotations from the timeline, terminal, and browser now share a single compact comment editor while keeping source highlights attached to local drafts.

— Behavioural description only, no surface named to act on v2.2.2

WAS THIS USEFUL? 👍 👎

planning-with-files

Feedback ⓘ v3.26.0 NOTES ↗ RANK

planning-with-files is a persistent file-based planning system for AI coding agents with crash-proof markdown plans, session recovery, and context-rot protection.

> planning-with-files added an environment variable to suppress plan context injection for Claude Code subagent tool calls, reducing noise in multi-step orchestrations.

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

01 **Suppress subagent plan context injection** NEW 85

Setting `PWF_SKIP_SUBAGENTS=1` suppresses plan context and progress reminders for Claude Code subagent tool calls, cutting noise in multi-step orchestrations without disabling planning globally. The feature fails closed: if the hook cannot classify its input, unrecognised calls still receive context.

Silence mid-task plan reminders for subagent tool calls in Claude Code to reduce noise in multi-step orchestrations without disabling planning globally.

```
$ PWF_SKIP_SUBAGENTS=1 claude
```

— Names exact env var, fail-closed behaviour, and a runnable command. v3.26.0

WAS THIS USEFUL? 👍 👎

coffee-cli

Feedback ⓘ v3.7.9 NOTES ↗ RANK

coffee-cli is a clean AI terminal workspace with quick input, folder switching, copy & paste, and image pasting.

> Coffee CLI v3.7.9 adds the ability to send tasks directly into active Claude or Codex desktop sessions and to resume conversations in a chosen Terminal or Desktop mode.

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

01 **Send tasks to Claude/Codex desktop sessions** NEW 55

Notes, lists, and categorized prompts can be sent to the current Claude or Codex desktop session; the status updates on success, and content is retained if the send fails.

— Describes behaviour and failure handling but no exact command or UI path v3.7.9

02 **Resume sessions in Terminal or Desktop mode** NEW 40

Coffee CLI can resume Claude Code and Codex conversations using the card's selected Terminal or Desktop mode, continuing the original conversation from where it left off.

— Explains the capability but lacks concrete steps or settings names v3.7.9

WAS THIS USEFUL? 👍 👎

juggler

Feedback ⓘ v0.7.9 NOTES ↗ RANK

The Juggler Code Agent

> Juggler v0.7.9 turns `juggler run` into a true headless mode with auto-continue on cutoffs and rate limits, adds per-project environment variables, an `/init` command that drafts AGENTS.md, and a six-language UI, alongside raising the open-window cap from 11 to 100 and smaller workflow improvements to forking, scheduling and cross-model reasoning.

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

01 **Headless `juggler run` mode improvements** IMPROVED 75

`juggler run` now refuses tools that wait for user input, automatically continues a reply cut off at Max output, and for usage-limit errors that include a reset time offers to auto-continue once the limit lifts.

Run a headless agent task without any user-interactive tools, letting juggler continue automatically if the reply is cut off at Max output.

```
$ juggler run 'Refactor the auth module and write tests'
```

— Mechanism explained and a runnable command example provided v0.7.9

02 **Custom endpoint headers and models.dev provider suggestions**

IMPROVED 75

Custom endpoint headers now accept `{{sessionId}}` as a per-conversation ID placeholder, and custom endpoints also suggest models.dev providers, auto-filling the URL and each model's limits.

Include a per-conversation ID in a custom endpoint header so your proxy can correlate requests back to a specific Juggler conversation.

```yaml
Authorization: Bearer {{sessionId}}
```

— Named placeholder and an example header a reader can copy v0.7.9

03 **`/init` command drafts AGENTS.md** NEW 60

A new `/init` command drafts an `AGENTS.md` file for the project, and projects without one are offered it.

— Exact command and file named, but no further mechanism v0.7.9

THINNER COVERAGE BELOW

04 **Multi-language interface support** NEW 50

The interface is now available in Japanese, Korean, German, French, Spanish and Brazilian Portuguese, selectable under Settings → Language.

— Languages and UI path given, nothing deeper to act on v0.7.9

05 **Cross-call reasoning visibility for more models** IMPROVED 50

GLM, DeepSeek, Kimi and local models now see their own earlier reasoning across tool calls.

— Named model families but no mechanism or config shown v0.7.9

06 **Ctrl+Tab conversation switching** NEW 50

Ctrl+Tab returns to the last-used conversation; holding Ctrl lets you pick from recent conversations.

— Exact keybinding given, directly usable as described v0.7.9

07 **Server-side scheduled sends** IMPROVED 45

Scheduled sends now fire from the server even with every window closed, and your own send-mode slash commands can be scheduled.

— Behavior explained but no config or command syntax given v0.7.9

08 **Increased open-window limit to 100** IMPROVED 40

Up to 100 windows can now be open at once, up from 11 previously (or 2 on Windows).

— Concrete before/after numbers but nothing to configure v0.7.9

09 **Fork conversation from any assistant reply** IMPROVED 40

You can now fork a new conversation after any assistant reply, not only at your own messages.

— Clear behavior change but no UI path or mechanism detail v0.7.9

10 **Per-project environment variables** NEW 35

Set per-project environment variables for every command using files or the Environment pin.

— No config file path or key named, just a brief description v0.7.9

WAS THIS USEFUL? 👍 👎

devoxxgenieideaplugin

Feedback ⓘ v1.15.0 NOTES ↗

DevoxxGenie is an agentic plugin for IntelliJ IDEA that uses local LLM's (Ollama, LMStudio, GPT4All, Jan and Llama.cpp) and Cloud based LLMs to help review, test, explain your project code. Latest version now also supports Spec Driven Development with CLI Runners.

DevoxxGenie adds favorite models, per-project Scan & Copy settings, and a Markdown-returning `fetch_page` agent tool with safety bounds.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.15.0
https://github.com/devoxx/DevoxxGenieIDEAPlugin.git
# already have the repo? check out this version:
$ git checkout v1.15.0
```

Mark any model as a favorite using the star next to the model dropdown; a new Favorites entry lists favorites across all providers, checks availability in the background, and survives restarts — remove favorites from the LLM settings page.

Store Scan & Copy settings — included file extensions, excluded folders and files, `.gitignore` handling, and RAG directory exclusions — per project in the project's workspace file instead of globally; projects without their own settings inherit the current global values.

The `fetch_page` agent tool now returns Markdown (headings, fenced code blocks with language, tables, lists, and absolute links) instead of flattened plain text, drops scripts/styles/ `nav` / `aside` / `footer` `/hidden` elements, prefers `main` when present, and enforces a 10 MB / 60 s cap with same-host-only redirects, URL validation, loopback/link-local rejection (e.g. `169.254.169.254`), and a 15-minute page cache.

Provider and model dropdowns are now hidden when they contain only one entry and reappear when there is a choice, reducing clutter.

NEW INTEGRATION Claude 5 family model support for Anthropic and Bedrock

UPDATED INTEGRATION MCP client migrated to langchain4j 1.20+ with Streamable HTTP transport

└──▷ BREAKING ON UPGRADE

! SSE-only MCP servers are no longer supported after the migration to the langchain4j 1.20+ MCP client, which removed the legacy 2024-11-05 HTTP+SSE transport.

WAS THIS USEFUL?  

Amazon Kiro

Feedback ⓘ

IDE 1.4.62: Memory Across Chat Sessions (2026-10-28) BLOG ↗ RANK

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

> Kiro's biggest addition is Memory, which carries repository conventions and past decisions across chat sessions, alongside a Windows system-wide installer, easier file drag-and-drop into chat, framed Mermaid diagrams, session reconnect without re-entering credentials, and keyboard-driven commenting in Markdown Review.

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

01 Keyboard-driven commenting in Markdown Review IMPROVED

88

Markdown Review now shows a Comment prompt when text is selected, allowing users to copy with `Cmd/Ctrl+C`, open a comment box with `Cmd+Option+M` or `Ctrl+Alt+M`, or start typing immediately. `Cmd/Ctrl+Enter` sends all comments to Kiro, while `Enter` saves a comment and starts a new line. The panel also now remembers whether it was left open or closed between sessions.

Review a Kiro-generated document and send comments without leaving the keyboard.

📍 In Markdown Review, select text › press Cmd+Option+M (or Ctrl+Alt+M on Windows/Linux) to open a comment › type your note › press Enter to save and start a new line › press Cmd+Enter (Ctrl+Enter) to send all comments.

— Every shortcut named, directly runnable IDE: Reconnect After Session Expiry and…

02 Memory across chat sessions NEW

82

Kiro now carries context — repository conventions, corrections, and approaches already ruled out — from one chat session to the next via Memory. The `Kiro Agent: Memory` setting controls how Kiro uses and learns from memories, including a `Use only` mode that lets Kiro read memories without writing new ones; the Memory view is opened via the Memories icon in the activity bar, and chat indicates when Kiro accesses or changes a memory.

Prevent Kiro from adding or modifying memories while still benefiting from existing ones — useful when working in a shared or regulated codebase.

📍 In Kiro IDE, go to Settings and set `Kiro Agent: Memory` to `Use only`. Kiro will read existing memories but make no changes.

Prevent Kiro from writing new memories mid-session while still letting it draw on existing ones — useful when you want stable, reviewed context only.

> 📍 In Kiro IDE, open Settings and navigate to Kiro Agent: Memory. Set the value to `Use only`. Kiro will read from the Memory view but will not add or change memories during chat.

— Names exact setting and mode plus UI location `IDE 1.4.62: Memory Across Chat Sessions…` `IDE 1.4.51: Memory Across Chat Sessions…`

---

03 **Reconnect after AWS session expiry** IMPROVED 65

When an IAM Identity Center or AWS Builder ID session expires, a new `Reconnect` button lets users resume sign-in in the browser without choosing a sign-in method or re-entering their Start URL and AWS Region.

When your AWS session expires mid-session, reconnect without re-entering credentials.

> 📍 In the IDE, when the session-expired prompt appears, select Reconnect › complete sign-in in your browser.

— Names exact sign-in flow skipped and UI control

`IDE: Reconnect After Session Expiry and…`

THINNER COVERAGE BELOW

04 **Framed Mermaid diagrams in chat** IMPROVED 57

Mermaid diagrams in chat now render in a framed block matching the editor theme, keeping arrows visible on dark themes. Diagrams scroll normally, or can be zoomed by holding Ctrl while scrolling (or pinching on a trackpad).

— Exact zoom interaction given, modest scope `IDE: Windows System-Wide Installer, Dra…`

---

05 **Windows system-wide installer** NEW 55

A new x64 or ARM64 installer makes Kiro available to every user on a Windows machine, requiring administrator rights to run.

— Names architectures but no command given `IDE: Windows System-Wide Installer, Dra…`

---

06 **Drag files into chat without Shift** IMPROVED 55

Files can now be dragged from the desktop or file manager directly into Kiro chat to attach them, no longer requiring the Shift key to be held; dropping an unsupported file type surfaces an error with the file name, or a count when multiple files are dropped.

— Describes behaviour and error case, no exact UI path

`IDE: Windows System-Wide Installer, Dra…`

07 **File save failure reporting** IMPROVED 35

Kiro now reports when a file edit fails to save, instead of incorrectly reporting that the save succeeded.

— Bare behaviour change, no mechanism or scope given

IDE: Reconnect After Session Expiry and…

WAS THIS USEFUL?  

forgedock

Feedback ⓘ v1.37.0 NOTES ↗ RANK

ForgeDock — autonomous AI dev pipeline for Claude Code. GitHub as agent memory: issue in, PR out.

> ForgeDock shipped a built-in compute pool and host scheduler for distributing lane execution, hybrid local/remote sandboxing, a native unattended milestone train with governance guard gates and anchored decisions, plus a wave of batch-orchestration, review, and CI-reliability improvements across ten releases.

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

01 **Built-in compute pool for lane execution** NEW 95

forgedock ships a built-in compute pool with `forgedock pool router|enroll|status|upgrade|remove` commands and a `pool` execution provider; pool agents run on Linux via a native systemd installer bundled under `bin/pool` and `pool/`; setting `execution.provider: pool` in `forge.yaml` routes lanes to the local pool router by default; a local fallback activates when the pool has no capacity while opted into `mode remote`, with agent liveness inspectable through the same setting; locking the lane image version means running `forgedock update` also triggers a pool upgrade; and per-agent remote lane minutes appear in the batch report with pool utilisation visible in `forgedock stats`.

Enroll a Linux machine as a pool agent so lanes can offload to it instead of running locally.

```
$ forgedock pool enroll
```

Route all lane execution to the local compute pool by setting the provider in your repo config.

```yaml
execution:
  provider: pool
```

Check pool agent health and utilisation after enrolling agents.

```
$ forgedock pool status
forgedock stats
```

— Full named subcommands, config key, and runnable examples. v1.33.0

02 **Deterministic staging review command** NEW 95

forgedock adds `forgedock review-staging <PR>` for a deterministic pre-dispatch staging review covering route, open-finding gate, scope, prepare, roster, and reviewer briefs; `forgedock review-staging --delta <sha>` reviews only the incremental commits since a previous pass while crediting the earlier review; and a post-dispatch run of `forgedock review-staging` handles delivery, extract/classify/file-prepare, and the Phase 8 gate marker.

Review only the commits added since a previous staging pass, crediting work already done, to avoid re-reviewing unchanged code.

```
$ forgedock review-staging --delta abc1234 987
```

Run a full deterministic staging review of a PR before dispatch, checking routing, open findings, scope, and reviewer briefs.

```
$ forgedock review-staging 987
```

— Runnable `review-staging` command with a delta-mode example. v1.31.0

03 **schedd, gc, bootstrap and limits CLI subcommands** NEW 95

forgedock adds `forgedock schedd` to run the host scheduler (orchestrates, submits jobs, and streams output, with `--resume` to reattach to a running scheduler); `forgedock gc` as a reaper to identify and clean up owned leftovers (dry-run by default, `--apply` to apply deletions, scoped to specific ownership boundaries); `npx forgedock bootstrap` to run headlessly (`--dry-run` for a plan preview, or resume from existing GitHub state); and `forgedock limits show` / `forgedock limits clear` to inspect or clear account-scoped usage-limit holds keyed by `backendIdentity`.

Start the host scheduler so multiple repos share one admission queue and restart-safe lane tracking across sessions.

```
$ forgedock schedd
```

Preview stale or orphaned resources before committing to deletion — then promote to a real run with `--apply`.

```
$ forgedock gc
forgedock gc --apply
```

— Multiple runnable subcommands and flags with usage examples. v1.28.0

04 **Host scheduler execution controls and fairness** NEW

90

forgedock's host scheduler gains `execution { mode, max_remote }` in `forge.yaml` to set execution mode and a remote-runner ceiling with ledger-tracked sandbox teardown; `FORGE_HOST_PARALLELISM` caps host-level verification concurrency across all lanes; phase-aware, reservation-based admission lets many cheap LLM-phase lanes run freely while heavy steps are semaphore-gated; cross-repo fairness weights, a host-global lane cap, and cross-batch priority admission (P0 first) govern scheduling; and the batch report now shows per-lane sandbox cost alongside a hard execution budget cap.

— Names `execution{mode,max_remote}`, `FORGE_HOST_PARALLELISM`, and scheduling policies.

v1.28.0

05 **Dynamic batch issue management: add, hold, release, resume**

NEW

88

forgedock lets operators requeue a terminal issue into a running batch with `forgedock orchestrate --add <id> N...` (skipping issues already in the batch), hold or release individual issues within a batch, auto re-admit skipped dependent issues once their blocker merges mid-batch, and use `--resume` to re-admit issues from a cancelled batch.

Requeue a terminal issue that failed in a running batch after investigating and deciding to retry it.

```
$ forgedock orchestrate --add 4550
```

Resume a cancelled batch, re-admitting any issues that were skipped or held, then run up to 3 lanes in parallel.

```
$ forgedock orchestrate 41 42 43 44 --parallel 3 --resume
```

Add a newly filed urgent issue to a batch that is already running rather than queuing a separate batch.

```
$ forgedock orchestrate --add 4501 4502 4503
```

— Runnable `orchestrate --add`/`--resume` commands with worked examples. v1.36.0 v1.35.0 v1.31.0

06 **Issue decomposition: budget scaling, gating, and parent auto-close**

NEW 83

forgedock auto-decomposes or scales budgets for L-sized issues when lanes hit `budget.exceeded` instead of stalling; gates decomposition on independence and size, falling back to a single checklist lane when neither threshold is met; and closes parent issues automatically after decomposition via the `decomposition.close_parent` config key, with closed decomposed parents still gating dependents through their sub-issues.

Control whether forgedock closes a parent issue automatically once all its decomposed sub-issues are created.

```yaml
decomposition:
  close_parent: true
```

— Names `budget.exceeded` and `decomposition.close_parent` with a config example. v1.35.0 v1.31.0 v1.30.0

07 **Hybrid local/remote sandbox execution** NEW 80

forgedock enables hybrid execution to keep cheap phases local and ship heavy phases to a remote sandbox, using a pluggable sandbox-provider adapter with a shared cache store and per-repo scoped secrets.

Configure hybrid remote execution so cheap LLM phases stay local while heavy build/test phases run in a remote sandbox, with a ceiling of 4 concurrent remote runners.

```yaml
execution:
  mode: hybrid
  max_remote: 4
```

— Config example sets `mode: hybrid` and `max_remote` directly. v1.28.0

08 **Native unattended milestone train** NEW 75

forgedock adds a native unattended milestone train so external chains no longer need to mutate `branches.staging` or duplicate milestone state; adds milestone dependency validation against live GitHub state that refuses to start parallel milestones with unmet dependencies; adds scheduled full-suite integration runs on active milestone branches; and models external prerequisites as preview/deploy gates while generating a readiness report.

— Names the `branches.staging` key it replaces but gives no direct command. v1.37.0

09 **forgedock init: empty repos and Projects v2 boards** IMPROVED

75

`forgedock init` now supports empty repos and repos with no remote, creating the repo with an App-installation-aware owner picker and a staging branch; it also automatically creates or links a GitHub Projects v2 board, populates it with ForgeDock fields, detects a missing project scope, and derives domain labels.

— Names the `forgedock init` command and its new setup behaviors. v1.27.0

10 **Guard gates and anchored decisions require supersedes** NEW

70

forgedock adds a guard gate that reruns base versions of tests guarding lines a lane changes, requiring an explicit `supersedes` amendment to proceed if a guarded test breaks, and adds anchored decisions that bind the paths they govern, where changing one requires a `supersedes` decision that updates the register.

— Mechanism explained via `supersedes`, but no command or flag given. v1.37.0

11 **Pre-push invariant checks** NEW

70

forgedock runs repo-declared invariant checks via `verify.invariants` before every push, so registry and coverage suites catch failures locally rather than only in CI.

— Names the `verify.invariants` config key that drives the check. v1.35.0

12 **`forgedock/verify` PR check run** NEW

70

forgedock posts deterministic gate results as an opt-in `forgedock/verify` check run on pull requests so teams can gate merges on conformance results without reading structured comments.

— Names the exact GitHub check run and its opt-in nature. v1.29.0

13 **refactor:split lane for symbol-based codemods** NEW

70

forgedock's new `refactor:split` lane groups symbols via a prompt over a deterministic symbol index, runs Python and TypeScript codemod movers, verifies no new import cycles, and routes by label.

— Names the lane and its movers but not the trigger command. v1.28.0

14 **Builder attempt trigger tagging and cost breakdown** NEW

65

forgedock tags every builder attempt with its trigger in `PHASE_ATTEMPT_USAGE` and lets you view cost broken down by trigger in `forgedock stats`.

— Names the `PHASE_ATTEMPT_USAGE` field and the `forgedock stats` command. v1.31.0

15 **Orchestrator health and hotspot detection** NEW 65

forgedock's orchestrator reserves a share of lanes for deterministically scanned health issues via a maintenance budget, detects hotspots at plan time to route new code out of over-threshold files and block unexempted growth, and generates a deterministic milestone drift summary from per-lane structure deltas, filing an issue only when a configurable threshold is crossed.

— Describes mechanisms but names no config key or command. v1.28.0

16 **Mechanical backlog plan validation** NEW 65

forgedock validates a backlog plan mechanically before any lanes run using `scripts/bootstrap-validate.mjs`.

Catch structural problems in a backlog plan before committing any lane budget to it.

```
$ node scripts/bootstrap-validate.mjs
```

— Gives a runnable validation script with an example. v1.27.0

17 **Post-deploy and date-gated follow-up queue** NEW 60

forgedock adds post-deploy and date-gated follow-up types that need no build lane, auto re-queue on promotion or on a date, and run a read-only verification lane; it also adds a single post-deploy checklist per release that classifies deferred rows at contract time and collapses per-lane follow-ups.

— Clear behavior described, no config or command surface named. v1.37.0

18 **Per-lane process resource caps** NEW 60

forgedock now caps every local lane's process tree by default using `TasksMax` and `MemoryMax` so one runaway lane cannot OOM the host.

— Names the limits applied but no configuration surface shown. v1.37.0

19 **Light lane execution mode** NEW 60

forgedock adds a light lane execution mode that runs a single builder session with a derived contract and deterministic gates, auto-upgrading to a full lane when needed; batch telemetry reports now track a `FORGE:LITE` receipt with a light/full lane split and upgrade rate.

— Explains the mechanism but gives no flag to select it. v1.32.0

20 **Parallel scoped pre-PR checks** IMPROVED 60

Pre-PR checks now run learned commands in parallel, scoped to changed files rather than the whole repo, cutting precheck waits that previously reached 14–25 minutes per lane.

— Gives before/after timing but no flag to control it. v1.30.0

21 **Autonomous mechanical failure recovery in orchestrate** NEW 60

forgedock's `orchestrate` command now auto-recovers mechanical `needs-human` failures in-batch, re-queues dependents, and escalates only on product, spend, or release decisions.

— Names the `needs-human` state and the escalation scope. v1.29.0

22 **Resource hygiene: doctor, preflight GC, cleanup** NEW 60

forgedock runs the `doctor` resource hygiene check, orchestrates preflight GC, and exposes a `/cleanup` unification surface via the new hygiene config.

— Names `doctor`, preflight GC, and `/cleanup` surfaces. v1.28.0

THINNER COVERAGE BELOW

23 **Deterministic public-API-shape change detection** NEW 55

forgedock detects public-API-shape changes deterministically — signature, sync↔async, generator↔function, and removed or renamed symbols.

— Lists detected change types but no command or output shown. v1.31.0

24 **Lane Docker resource tracking and teardown ledger** NEW 55

forgedock tracks all Docker resources created by lanes and tests with `forgedock.*` ownership labels, and adds a lane resource ledger that guarantees teardown on every exit path, including crashes.

— Names the `forgedock.*` label convention but no inspection command. v1.27.0

25 **Signed outbound lifecycle event stream** NEW 50

forgedock can stream versioned outbound lifecycle events to a JSONL sink or a signed webhook for downstream consumption.

— Names sink formats but no config key or endpoint given. v1.36.0

26 **Fail-soft rate alerting** NEW 50

forgedock alerts on degraded fail-soft rates — deps cache, deferred-to-ci, toolchain warnings, and outcome spikes — via the stats subsystem.

— Names tracked signals but no command or threshold given. v1.31.0

27 **Structured FORGE:FACTS context block** NEW 50

forgedock emits a structured `FORGE:FACTS` block of decisions, rejected approaches, and hot spots that later lanes can read.

— Names the block but no access mechanism is given. v1.31.0

28 **Greenfield milestone bootstrapping** NEW 50

forgedock accepts a precomputed `/bootstrap` plan in milestone and decompose flows to skip code-read gating for greenfield milestones, and accepts planned Affected Files validated against a declared repo layout in greenfield mode for new issues.

— Names the `/bootstrap` plan surface but no command shown. v1.28.0

29 **Compounding-defect stats** NEW 45

forgedock adds compounding-defect stats measuring reintroduction rate, reopen-after-close, guard saves, and decision churn, all trended by repo age.

— Named metrics but no command or dashboard path given. v1.37.0

30 **Production probes with outcome-based acceptance** NEW 45

forgedock runs declared production probes post-promotion and reopens the issue on failure with outcome-based acceptance.

— Names the behavior but not how probes are declared. v1.31.0

31 **Test-map driven pre-PR check selection** NEW 45

forgedock feeds a test-map and repo-facts file into pre-PR check selection so the engine picks the relevant checks before a pull request opens.

— Names inputs used but no file path or command given. v1.29.0

32 **Preserve rows promoted to contract rows** IMPROVED 40

forgedock now promotes engine-authored preserve rows — guards and decisions on deliverable paths — to contract rows rather than prompt text.

— Thin description of an internal representation change. v1.37.0

33 **Budgeted context pack from issue graph** NEW 40

forgedock compiles each issue's GitHub graph into a budgeted context pack that converts issues into model prompts, with shadow measurement of context size.

— Describes the mechanism but no size limit or config given. v1.29.0

34 **Shadow-first deterministic review triggering** NEW 40

forgedock enables deterministic review triggering with shadow-first mode — the LLM reviewer is skipped only off low-risk paths.

— One-line mechanism, no config or threshold named. v1.28.0

35 **Stale review context detection** IMPROVED 35

forgedock's review step now detects and refreshes a stale `review.context` and `tech_stack`.

— One-line behavior change, no mechanism detail given. v1.37.0

36 **Risk-based builder model routing** NEW 35

forgedock routes the builder model by issue size and risk class with in-lane escalation.

— One-line behavior with no routing criteria or command. v1.31.0

37 **Independent adversarial verifier pass** NEW 35

forgedock runs an independent fresh-context, different-model verifier with an adversarial pass.

— Thin one-line description of the verifier. v1.31.0

38 **CI-runner lane verification** IMPROVED 30

forgedock runs lane verification on CI runners instead of the host to catch deploy-PR failures before merge.

— One-line description, no config or command given. v1.27.0

WAS THIS USEFUL? 👍 👎

octo-agent

Feedback ⓘ v1.17.21 NOTES ↗ RANK

Open-source, single-binary, self-hosted AI agent — your models and data stay on your machine. A coding agent on par with Claude Code and a personal assistant lighter than OpenClaw, across eight interfaces (TUI, CLI, Web, desktop, IM, VS Code, Obsidian, Mobile), all in one MIT-licensed Go binary running on any model.

> octo-agent's biggest push this window was turning its mobile app into a full-featured client — chat attachments, GenUI panels, session and task management, skills/MCP server CRUD, and a consolidated Me tab — alongside MCP cancellation/progress notifications and configurable concurrency caps for sub-agents and web requests.

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

01 Mobile chat interface: attachments, editing, and session context

NEW 75

Mobile app chat now supports attaching images and files, editing a sent message, rendering GenUI (`octo-ui`) panels, showing context usage and supporting context compacting, displaying sub-agent and workflow runs, showing the session goal, showing the session plan and turn progress, and showing tool call details.

— Enumerates every chat capability added but gives no UI path. v1.17.19

02 Mobile session and task management NEW 75

Mobile app now supports creating, editing, and deleting scheduled tasks; renaming, deleting, and copying sessions from the long-press sheet; switching model and permission mode inside a session; pinning sessions to keep them at the top of the list; and showing projects alongside individual sessions in the Sessions feed.

— Lists every session action but no exact UI command. v1.17.19 v1.17.18

03 MCP cancellation and progress notifications NEW 70

octo-agent's MCP layer now sends `notifications/cancelled` for abandoned requests so downstream MCP clients can clean up reliably, and shows tool-call progress reports that can extend the active request's timeout.

— Names the exact MCP notification type sent. v1.17.18

04 Mobile Me tab consolidation and profile features NEW 70

The mobile app now merges Config and Settings into a single Me tab, which shows agent profiles, the soul/user profile, and read-only endpoints; adds a per-device UI language

switch; lets users read and forget memories and restore or delete trashed files; and only shows the connection banner when the app is disconnected instead of always.

— Names every Me tab capability but no field-level config. v1.17.19 v1.17.18

05 **Mobile app Light Apps, session artifacts, and direct server connection**

NEW 65

The mobile app now supports Light Apps and session artifacts on the phone, and connects straight to `octo serve` without manual configuration.

— Names the `octo serve` connection but not Light Apps mechanism. v1.17.17

THINNER COVERAGE BELOW

06 **Skills and MCP server management on mobile** NEW 55

Mobile app now supports creating, editing, using, importing, and deleting skills and MCP servers.

— Describes CRUD scope but no config file or command referenced. v1.17.19

07 **Test saved endpoints by id** NEW 45

octo-agent's server now supports testing a saved endpoint by id.

— No endpoint path or method given, just the capability name. v1.17.19

08 **Configurable concurrency caps for parallel operations** NEW 40

octo-agent now lets users set configurable caps on sub-agent and web request concurrency, controlling how many parallel operations octo runs at once.

— No named config key or flag given for the caps. v1.17.21

09 **Signed Android APKs attached to releases** NEW 40

Signed Android APKs are now automatically attached to every octo-agent GitHub release.

— Simple release artifact addition with no further mechanism. v1.17.17

NEW INTEGRATION Kimi K3.1 Pro provider

RETIRED INTEGRATION Kimi K2.6 and K2.7-code from Kimi Coding Plan catalogue

v1.17.20 NOTES ↗ STABLE

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.17.20 https://github.com/open-octo/octo-agent.git
# already have the repo? check out this version:
$ git checkout v1.17.20
```

└──▷ **BREAKING ON UPGRADE**

! Models `kimi-k2.6` and `kimi-k2.7-code` are removed from the Kimi Coding Plan catalogue; configurations referencing them will break on upgrade.

WAS THIS USEFUL? 👍 👎

◆ Model Training & Fine-Tuning

NVIDIA Megatron Bridge

Feedback ⓘ v0.7.0 NOTES ↗

RANK

Megatron Bridge connects Hugging Face model workflows with Megatron Core for distributed model training and checkpoint conversion.

> Megatron Bridge shipped day-0 support for K-EXAONE-2 (750B-A37B MoE) and GLM-5.2's cuDNN fused DSA for 128K context-parallel training, alongside a broad sweep of new model integrations, new checkpoint conversion/export paths (MXFP8, Nemotron Omni MIMO, Qwen3.5 MoE+MTP), and training infrastructure additions like global-batch online packing and GTP rematerialization controls.

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

01 **New model architecture support** NEW 80

Adds day-0 support for `K-EXAONE-2` (750B-A37B MoE) from LG AI Research on the `k-exaone-2` branch with checkpoint conversion, inference, SFT/PEFT (LoRA) fine-tuning, and an NVFP4 inference-optimized checkpoint; adds `Nemotron 3.5 Super VL 120B-A12B` vision-language model support with HF-to-Megatron checkpoint conversion, image inference, pretraining, SFT, and LoRA recipes plus a new text-only checkpoint conversion path; and expands model coverage with `BAGEL-7B-MoT` (requires the `bagel` extra), `Gemma 4` (26B-A4B MoE and 31B dense) and `Gemma 4-VL` (26B-A4B MoE), `Kimi K2`, `Kimi-K2.5-VL`, and `Kimi K3`, `MiniMax-M2`, `M2.5`, `M2.7`, and `M3`, `Gemma 3-VL` and `Gemma 3`, `Ministral 3` (3B, 8B, and 14B), `Muse Glimmer 30B`, `DeepSeek V4` and `V4 Flash` (alongside existing `DeepSeek V3`), and `BERT`.

— Names branch, fine-tuning paths and checkpoint format Many named models and sizes but little mechanism per model Lists conversion path and training stages supported

v0.7.0

02 **New checkpoint conversion and export capabilities** NEW 70

Adds new checkpoint conversion and export capabilities: MXFP8 parameter export via `feat(ckpt): support MXFP8 parameter export`, `Nemotron Omni` `MIMO` conversion support, and `Qwen3.5` MoE text checkpoint extraction with `MTP` via conversion tooling.

— Names exact commit-style feature identifiers for each capability v0.7.0

03 **New training recipes for Qwen3.5 and Nemotron 3 Ultra** NEW 60

Adds `Qwen3.5` long-context SFT recipes — a text `MXFP8` recipe and `Qwen3.5 VL GB200` recipes — plus a Nemotron 3 Ultra 256-GPU VR200 NVFP4 performance recipe.

— Names specific recipes, hardware scale and precision formats v0.7.0

THINNER COVERAGE BELOW

04 **GLM-5.2 cuDNN fused DSA for 128K context training** NEW 50

Megatron Bridge now supports GLM-5.2 with cuDNN fused DSA for 128K THD-packed context-parallel training.

— Names mechanism and context length but no usage path product docs

05 **GTP rematerialization runtime controls** NEW 50

Control GTP rematerialization at runtime with newly exposed runtime controls via `feat(training): expose GTP rematerialization runtime controls`.

— Named feature identifier but no exact flag given v0.7.0

06 **Global-batch online packing during training** NEW 45

Enable global-batch online packing during training using Megatron-Core's sequence-packing scheduler.

— Names the scheduler but not a config entry point v0.7.0

07 **Independent vision recomputation config for Qwen** IMPROVED 40

Configure vision recomputation independently of the decoder for Qwen models.

— States the capability but no config key or flag v0.7.0

08 **Custom Energon task encoder configs** NEW 30

Support custom Energon task encoder configs for data pipelines.

— Names config target but no keys or examples v0.7.0

09 **Nemotron-H router replay diagnostic** NEW 25

Add a Nemotron-H router replay diagnostic for checkpoint debugging.

— Bare description with no mechanism or usage detail v0.7.0

WAS THIS USEFUL? 👍 👎

DeepSpeed

Feedback ⓘ v0.19.9 NOTES ↗ RANK

DeepSpeed is a deep learning optimization library for distributed training and inference, providing ZeRO sharding, pipeline parallelism, and mixed-precision execution.

> DeepSpeed's v0.19.9 release adds new Triton-based kernels for distributed loss computations and expert activations, alongside an RVV-vectorized CPU Adam kernel for RISC-V hardware.

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

01 **RVV vectorized FP32 CPU Adam kernel** NEW 40

DeepSpeed adds an RVV (RISC-V Vector) vectorized FP32 kernel to CPU Adam, enabling hardware-accelerated optimizer steps on RVV-capable CPUs.

— Explains purpose and target hardware but no config or flag to invoke it v0.19.9

02 **Triton kernels for vocab-parallel losses and expert activation**

NEW 35

New Triton kernels were added for vocab-parallel logprobs, entropy and KL divergence computations, plus a separate fused Triton kernel for clamped expert activation.

— Names the operations but no API, flag or usage detail given v0.19.9

WAS THIS USEFUL? 👍 👎

NVIDIA Megatron-LM

Feedback ⓘ core_v0.20.0 NOTES ↗ RANK

Megatron-LM trains transformer models at scale with tensor, pipeline, sequence, and context parallelism, and ships Megatron Core as a reusable library.

> Megatron-LM's core_v0.20.0 release adds DeepSeek V4 support, an experimental Megatron-FSDP v2 integration, prefill/decode disaggregation, and OpenTelemetry-based training observability, alongside new parallelism, MoE communication, and hybrid-architecture layer options.

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

01 **Megatron-FSDP v2 integration** NEW 83

Adds an experimental Megatron-FSDP v2 integration with a Megatron Core adapter and Megatron Bridge supporting context, expert, pipeline, and hybrid data-parallel sharding (the adapter requires `TP=1` and expert `TP=1`). Adds PyTorch distributed checkpoint save/load for model and optimizer state under Megatron-FSDP v2 (optimizer checkpointing currently excludes expert-parallel layouts), hybrid FSDP with configurable ZeRO levels per NVLink domain and separate configs for dense and MoE sections, CUDA graph capture and replay for supported dense-model configurations, and MXFP8 compute and quantized parameter gathers through the adapter alongside mixed-precision optimizer support.

— Rich mechanism and constraints given, but no exact CLI flags core_v0.20.0

02 **Prefill/decode disaggregation** NEW 60

Adds prefill/decode disaggregation with NCCL and NIXL KV-cache transfer paths and SSM-state handoff primitives.

— Names transfer paths and primitives, no setup instructions core_v0.20.0

03 **Expert-parallel communication backends** NEW 60

Adds NCCL EP eager mode alongside the existing full-iteration graph mode, adds MXFP8 dispatch-forward and combine-backward communication to the NCCL EP backend (experimental), and adds an experimental DeepEP v2 backend for the flex token dispatcher.

— Groups three EP backend additions, names each mode core_v0.20.0

THINNER COVERAGE BELOW

04 **ShortcutMoE routing in HybridStack** NEW 55

Adds ShortcutMoE shortcut-connected routing in HybridStack, overlapping expert all-to-all communication with paired attention or Mamba computation.

– Explains the overlap mechanism but not how to enable it core_v0.20.0

05 **OpenTelemetry training observability** NEW 52

Adds optional OpenTelemetry traces and metrics covering training, pipeline communication, gradient synchronization, and checkpointing.

– Names the four instrumented areas but no config flag to enable it core_v0.20.0

06 **DeepSeek V4 support via HybridModel** NEW 50

Megatron-LM now supports DeepSeek V4 via the `HybridModel` class.

– Names the class but gives no further mechanism core_v0.20.0

07 **Generalized Tensor Parallelism (GTP)** NEW 50

Adds Generalized Tensor Parallelism (GTP) with configurable batch/sequence partitioning, opt-in latent-projection sharding, and MTP support.

– Describes capabilities but no config keys or usage path core_v0.20.0

08 **Manifold-constrained hyper-connections (mHC)** NEW 50

Adds manifold-constrained hyper-connections (mHC) to `HybridModel`, including pipeline and virtual-pipeline parallel support.

– Names the feature and parallel support, lacks usage detail core_v0.20.0

09 **Gated Delta Product (GDP) layers** NEW 30

Adds Gated Delta Product (GDP) layers and model-building support for hybrid architectures.

– Only a bare name with minimal description core_v0.20.0

WAS THIS USEFUL? 👍 👎

Hugging Face TRL

Feedback ⓘ v1.17.0 NOTES ↗ RANK

TRL trains transformer language models with reinforcement learning, covering supervised fine-tuning, reward modeling, PPO, DPO, and GRPO.

> TRL v1.17.0 adds single-node fine-tuning for the 1T-parameter Mistral Large 4, AMD ROCm GPU support, and cross-tokenizer distillation in GOLDTrainer, while overhauling GRPO's vLLM importance sampling with new defaults and raising several minimum dependency versions as breaking changes.

└──▷ **WHAT SHIPPED · 12 FEATURES** most completely described first ? what's the number?

01 Single-node fine-tuning of Mistral Large 4 (1T params) NEW

95

SFTTrainer can now fine-tune `mistralai/Mistral-Large-4-1T-A52B` (1T parameters, 52B active) on a single 8×B300 node using DistributedConfig(fsdp_size=8, ep_size=8) for expert sharding, combined with `packing=True` and `assistant_only_loss=True`. Requires `transformers>=5.22.0`, `peft>=0.21.0`, and `accelerate>=1.15.0`.

Fine-tune Mistral Large 4 (1T params) on a single 8×B300 node with LoRA, FP8 weights, sequence packing, and assistant-only loss.

```python
from trl import SFTTrainer, SFTConfig
from peft import LoraConfig
from transformers import DistributedConfig

training_args = SFTConfig(
    model_init_kwargs={
        "dtype": "auto",
        "distributed_config": DistributedConfig(fsdp_size=8,
ep_size=8),
    },
    packing=True,
    assistant_only_loss=True,
    output_dir="mistral-large-4-lora",
)
trainer = SFTTrainer(
    model="mistralai/Mistral-Large-4-1T-A52B",
    args=training_args,
    peft_config=LoraConfig(...),
    train_dataset=dataset,
)
trainer.train()
```

— Full runnable config, exact classes and version floors given `v1.17.0`

02 Unbiased vLLM importance sampling in GRPO

BREAKING

93

GRPO and AsyncGRPO now renormalize log-probabilities over the `top_k` / `top_p` / `min_p` candidate set, following the Keep Sampling Mask method, so the vLLM importance sampling ratio is no longer biased toward 0. `GRPOConfig` now raises an error when `top_p < 1.0` or `min_p > 0.0` is set without `top_k > 0` under vLLM importance sampling correction — set `top_k` (e.g. `top_k=512`) or `vllm_importance_sampling_correction=False` to avoid it. AsyncGRPO also trains with score centering on by default (`score_centering=True`, `importance_sampling_min=0.5`, `importance_sampling_max=5.0`), dropping tokens whose ratio to the sampler falls outside `[0.5, 5.0]`; set `score_centering=False` to restore the prior clipped objective.

Run GRPO with vLLM top-p sampling without triggering the new importance-sampling validation error.

```python
from trl import GRPOTrainer, GRPOConfig

training_args = GRPOConfig(
    use_vllm=True,
    top_p=0.9,
    top_k=512,
    output_dir="grpo-output",
)
trainer = GRPOTrainer(
    model="Qwen/Qwen2.5-0.5B-Instruct",
    reward_funcs=reward_fn,
    args=training_args,
    train_dataset=dataset,
)
trainer.train()
```

— Exact config keys, defaults and migration path given `v1.17.0`

03 **Cross-tokenizer distillation in GOLDTrainer** NEW 86

`GOLDTrainer` adds X-Token cross-tokenizer distillation: set `lmbda=0.0` and `xtoken_loss_type='p_kl'` or `'h_kl'` with a precomputed projection matrix to run off-policy knowledge distillation between a student and teacher that use different tokenizers.

Use X-Token cross-tokenizer distillation in GOLDTrainer when the student and teacher use different tokenizers.

```python
from trl import GOLDTrainer, GOLDConfig

training_args = GOLDConfig(
    lmbda=0.0,
    xtoken_loss_type="p_kl",
    output_dir="gold-xtoken",
)
trainer = GOLDTrainer(
    model="student-model",
    args=training_args,
    train_dataset=dataset,
)
```

— Exact config keys and values with runnable example `v1.17.0`

04 **Minimum dependency version bumps** 75

BREAKING

TRL now requires `torch>=2.6.0` (dropping pre-2.6 FSDP2 and accelerator fallbacks), `accelerate>=1.11.0` (dropping the FSDP2 ignored-modules fallback), `peft>=0.17.0` (dropping `target_parameters` guards), `bitsandbytes>=0.45.2`, and `math-verify>=0.8.0`.

— Exact version floors and dropped fallbacks named v1.17.0

---

05 **Tree packing for Gated DeltaNet hybrid models** NEW

63

Tree packing now supports Gated DeltaNet linear-attention hybrids — Qwen3.5, Qwen3.6, Qwen3-Next, and OLMo-Hybrid — routing causal convolution and delta-rule recurrence through FLA's chunked kernels.

— Names models and kernel mechanism, no usage example v1.17.0

THINNER COVERAGE BELOW

06 **Model-object and LoRA flexibility for async trainers** IMPROVED

50

AsyncGRPO and `AsyncDistillationTrainer` now accept instantiated model objects in addition to model name strings; `AsyncDistillationTrainer` also gains MoE aux loss support and injected-worker LoRA name support.

— Grouped thin additions to the same async trainers v1.17.0

---

07 **Dynamic sampling in AsyncGRPO (DAPO)** NEW

45

AsyncGRPO now filters zero-advantage groups following DAPO's dynamic sampling, skipping forward and backward passes for groups where all rollouts received the same reward or none could be scored.

— Explains mechanism but no config flag is named v1.17.0

---

08 **AMD ROCm GPU support** NEW

35

TRL's trainers now run on AMD Instinct GPUs through PyTorch's ROCm build, with an accompanying guide and a CI job that runs the test suite on every PR.

— No install steps or hardware specifics given v1.17.0

---

09 **Mixed-precision checkpoint saving for async trainers** IMPROVED

30

AsyncGRPO and AsyncDistillation checkpoints are now saved in the mixed-precision dtype instead of fp32.

— States the change but no size or perf numbers v1.17.0

---

10 **vLLM 0.32.0 support** IMPROVED

25

TRL now supports vLLM 0.32.0, including patch releases.

— Bare compatibility note, no further detail v1.17.0

11 **Parquet logging for completions** IMPROVED 25

Logged completions are now written to parquet as JSON.

— No field names or schema given v1.17.0

12 **Removal of experimental TPOTrainer** BREAKING 25

The experimental TPOTrainer has been removed.

— No migration guidance given beyond removal v1.17.0

└──▷ **BREAKING ON UPGRADE**

! GRPOConfig now raises when `top_p < 1.0` or `min_p > 0.0` is set without `top_k > 0` when using vLLM with importance sampling correction. Set `top_k` (e.g. `top_k=512`) or set `vllm_importance_sampling_correction=False`.

! AsyncGRPO now trains with score centering by default (`score_centering=True`, `importance_sampling_min=0.5`, `importance_sampling_max=5.0`), dropping tokens whose ratio to the sampler falls outside `[0.5, 5.0]`. Set `score_centering=False` to restore the prior clipped objective.

! The experimental TPOTrainer has been removed.

! `torch>=2.6.0` is now required; pre-2.6 FSDP2 and accelerator fallbacks have been dropped.

! `accelerate>=1.11.0` is now required; the FSDP2 ignored-modules fallback has been dropped.

! `peft>=0.17.0` is now required; the `target_parameters` guards have been dropped.

! `bitsandbytes>=0.45.2` is now required.

! `math-verify>=0.8.0` is now required.

WAS THIS USEFUL?  

fizgig

Feedback ⓘ v8.9.95 NOTES ↗ RANK

Fizgig — LoRA & Fine-tune Studio for Klein 9B, Krea 2, MiniMax H3, Qwen Image 2.1, Anima & Z-Image Turbo and Ace-Step XL: train, fine-tune, profile, repair and extract LoRAs & LoKR for Image, Video & Music Models

> Fizgig tunes MOSS-Image's default epoch-sharing schedule for better likeness retention, adds reusable prompt sets to the Samples tab, and lets LoRA Royale score likeness against multiple reference photos.

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

01 **Prompt sets in Samples tab** NEW 75

Fizgig adds prompt sets under the Samples tab: use Save as set... to name a prompt collection and Load to apply it. Sets are shared across all models, and with Change the previews of the run going now ticked, a loaded set takes effect from the next preview of a live run.

— Exact UI control names given, clear navigation path. v8.9.94

02 **MOSS-Image Clean-end share default schedule** IMPROVED 73

MOSS-Image now trains with a new Clean-end share default of `60:7:30:8:3` for LoRAs and sliders — 60% until epoch 7, 30% at epoch 8, then 60% with every 3rd epoch at 30% — holding likeness about 5 points higher from epoch 9 to 20 than the previous default. Alternating epochs now train at the higher share by default, with every Nth epoch running at the lower share and half the learning rate to preserve likeness while settling composition.

— Exact ratio and epoch mechanism given, but it's a default, not a flag. v8.9.94

THINNER COVERAGE BELOW

03 **Multi-photo scoring in LoRA Royale** IMPROVED 55

LoRA Royale now scores likeness against up to 3 subject photos chosen from a thumbnail grid across any folder, computing the mean per epoch the same way the Samples gallery does.

— Explains mechanism and UI entry point but limited numeric detail. v8.9.94

NEW SUPPORTED PLATFORM Z-Image Base model support

NEW INTEGRATION Qwen3-VL 8B secondary captioning model

> v8.9.95 NOTES ↗ LATEST STABLE

Fizgig adds Z-Image Base as a standalone training model and a secondary Qwen3-VL 8B captioning model selectable in Preferences.

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/shootthesound/fizgig:8.9.95
```

git

```
$ git clone --branch v8.9.95
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v8.9.95
```

published under the project's GitHub account · checked 30 Oct **Wrong command?**

Z-Image Base is now a standalone model on the Training tab, supporting LoRAs, LoKR, sliders, masked training, and full fine-tunes with the same memory settings as Z-Image Turbo.

Training samples, Studio, and LoRA Royale preview Base LoRAs on Z-Image Turbo by default; untick `Use Z-Image Turbo for samples` on the Samples tab to preview on Base instead.

Preferences adds a `Captioning model` card with an optional `Secondary captioning model` slot for Qwen3-VL 8B (fp8 or bf16), covering the Captions tab and trainer auto-recaption.

WAS THIS USEFUL? 👍 👎

PyTorch TorchTitan

Feedback ⓘ v0.4.0 NOTES ↗

TorchTitan provides PyTorch-native model implementations and recipes for large-scale distributed training of generative models.

TorchTitan adds `--module` / `--config` recipe selection, `torch_remat` regional activation control, `model.local_compile_regions` for per-function kernel fusion, native HF safetensors loading, SDC replay for silent-data-corruption detection, a Grain data pipeline with checkpointable state, batch dimension folding, and a promoted TitanRL stack in `torchtitan.rl` with `ActorSystem` and `GenerationEngine`.

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install torchtitan==0.4.0
# already installed? upgrade to this version:
$ pipx install --force torchtitan==0.4.0
```

git

```
$ git clone --branch v0.4.0 https://github.com/pytorch/torchtitan.git
# already have the repo? check out this version:
$ git checkout v0.4.0
```

PyPI attestation · checked 30 Oct Wrong command?

Select Python recipe modules and functions with `--module` and `--config` on the CLI, moving general overrides into the recipe function itself.

Control activation saving and recomputation by region using `torch_remat`.

Compile individual functions such as SwiGLU, RoPE, and gated RMSNorm independently for kernel fusion using `model.local_compile_regions` instead of whole-TransformerBlock compilation.

NEW SUPPORTED PLATFORM Use `HiMidLoLinear` for BF16 matrix multiplications with FP32 outputs, combining two or three BF16 gradient parts to improve accuracy over a single BF16 cast.

TitanRL moves from `torchtitan.experiments.rl` to `torchtitan.rl`, sharing the core `TrainingEngine` and adding pipeline and context parallel training support.

+9 more

NEW SUPPORTED PLATFORM Muse Spark Open 816B-A42B

NEW SUPPORTED PLATFORM Kimi K3

NEW SUPPORTED PLATFORM DeepSeek V4

NEW SUPPORTED PLATFORM NVFP4 support for grouped MoE linear layers

└──▷ **BREAKING ON UPGRADE**

- ! Recipes must now be specified as fully qualified Python modules using `--module` and `--config`; general CLI overrides must move into the recipe function.
- ! Whole-TransformerBlock compilation in the standard trainer is removed; use `model.local_compile_regions` instead.
- ! Legacy Float8 training, MinimalAsyncEP, and the `full_dtensor` backend have been removed.
- ! TitanRL is now in `torchtitan.rl`; the former path `torchtitan.experiments.rl` no longer exists.
- ! MXFP8 training requires a TorchAO nightly build; TorchAO 0.18 does not include the required kernels.
- ! Model configurations must be rebuilt when changing sequence or expert parallelism.

WAS THIS USEFUL? 👍 👎

AGENT

◆ MCP Servers & Tooling

kordoc

Feedback ⓘ v4.24.25 NOTES ↗ RANK

모두 파싱해버리겠다 — HWP·HWPX·PDF·Office 문서를 Markdown으로. 양식 자동 채우기와 신구대조를 갖춘 CLI·MCP 서버 | Convert Korean documents (HWP, HWPX, PDF, Office) to Markdown — CLI and MCP server with form filling and diff

> kordoc's latest releases add resilience to document editing and rendering: patches now fall back to a rewritten file when in-place edits fail, encrypted HWPX files can be rendered with a password, and OCR gains a selectable medium-accuracy detection model for degraded scans.

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

01 Selectable medium OCR detection model NEW

88

OCR can now use the PP-OCRv6 medium detection model (~62 MB, SHA-256 pinned, downloaded on first use) for degraded inputs — skewed, blurred, scanned, or noisy documents — via `--ocr-det medium` on the CLI or `ocr_det: "medium"` in MCP; the default remains the v5 mobile model.

Run OCR on a scanned or skewed PDF using the medium detection model instead of the default mobile model to reduce character error rate on degraded input.

```
$ npx kordoc 스캔본.pdf --ocr --ocr-det medium -o 스캔본.md
```

— Names flag, MCP key, model size, hashing, and default behaviour v4.24.17

02 Password support for encrypted HWPX rendering NEW

78

kordoc can now render encrypted HWPX files by supplying a password via the `password` parameter on MCP `render_document` or the `--password` flag on CLI `render`, and it reports when a password is required instead of failing silently.

Render a password-protected HWPX file to preview its content without needing Hangul Office.

```
$ npx kordoc render 암호문서.hwpx --password <password> -o 미리보기.pdf
```

— Names both CLI flag and MCP parameter with a runnable example v4.24.25

03 Fallback rewrite output for patch command NEW

73

When `npx kordoc patch` cannot apply some edits in place, kordoc now also writes a fully formatted fallback rewrite file alongside the normal output, e.g. patching `원본.hwpx` with `편집.md` to `결과.hwpx` additionally produces `결과.rewrite.hwpx` if any edits could not be applied.

Patch a document and automatically receive a fully formatted fallback rewrite when some edits cannot be applied in place.

```
$ npx kordoc patch 원본.hwpx 편집.md -o 결과.hwpx
# If any edits could not be applied, kordoc also writes 결
과.rewrite.hwpx
```

— Runnable command shown but no detail on which edits trigger fallback `v4.24.25`

WAS THIS USEFUL? 👍 👎

memwhale

Feedback ⓘ v0.24.0 NOTES ↗ RANK

Persistent, local memory for developers and their coding agents. Records commands, output, errors, and the fixes that worked into SQLite and serves them over MCP.

> memwhale shipped an OpenHands SDK integration with its own test harness, a read-only pstack playbook for querying lessons and fixes, and two breaking changes to how command memories are scored and displayed in search.

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

01 OpenHands SDK integration and test harness NEW

88

memwhale adds an OpenHands SDK harness at `scripts/verification/openhands-mcp/` that runs `mw-mcp` under the real `openhands-sdk` 1.58.2 against temporary synthetic stores, covering discovery, fix handoff, and isolation across 12 tests. Configuring the OpenHands `MCPServer.env` requires setting `MEMORYWHALE_DATA_DIR` explicitly, since the server does not inherit it and `mw-mcp` otherwise opens the default store under `HOME`; the live check is run by setting `MW_OPENHANDS_LIVE=1` and `LLM_MODEL` before executing `live_check.py` in the same directory.

Point `mw-mcp` at a specific store when configuring an OpenHands MCPServer — without this the server opens the default store under HOME.

env

```
MEMORYWHALE_DATA_DIR=/home/user/.local/share/memorywhale/myproject
```

Run the OpenHands live check against a real model to verify the full `mw-mcp` integration end-to-end.

```
$ MW_OPENHANDS_LIVE=1 LLM_MODEL=gpt-4o python3
scripts/verification/openhands-mcp/live_check.py
```

— Names env vars, paths, version, and runnable commands v0.24.0

02 Keyword-similarity floor for filtered search BREAKING

71

Filtered searches — `mw search` with a project, or MCP's `search_memory` with `mode: "failures"` or `"lessons"` — now guarantee at least plain word-overlap keyword similarity for matching rows, via `memorywhale-core` 0.11.0's

`sqlite::salient_error_line`, so genuine matches no longer score 0%. In small or filtered result sets this can raise scores for rows that previously scored lower.

— Names exact modes, function, and version but no usage walkthrough v0.24.0

THINNER COVERAGE BELOW

03 **pstack playbook for memory recall** NEW 50

memwhale adds a read-only pstack playbook at `integrations/pstack/`, written in pstack's own format and filed under error tracking, so pstack's `why` and `recall` skills can query `mw-mcp` for past lessons and prior fix attempts.

— Named integration path but no runnable example given v0.24.0

04 **Error line leads search result text** BREAKING 50

Search results for failing commands now lead with the error line (the first line naming an error) rather than a banner or rule, consistently across the CLI, TUI, MCP, dashboard, and desktop app. This reorders the `text` field of a command memory, so code parsing that text for stderr lines will see a different order.

— Describes before/after but no field example or migration step v0.24.0

└──▷ **BREAKING ON UPGRADE**

! A command memory's `text` field now lists its error line first; code that parses that text sees stderr lines reordered.

! In small or filtered result sets, keyword similarity scores can rise: for a matching row, `memorywhale-core` 0.11.0's `sqlite::salient_error_line` ensures the score is never lower than plain word overlap.

WAS THIS USEFUL?  

chap

Feedback ⓘ v0.3.4 NOTES ↗ RANK

CHAP, the Collaborative Human Agent Protocol, is a MCP/A2A-compatible runtime for auditable human-agent work: approvals, overrides, handoffs, escalation and verifiable evidence logs.

> CHAP 0.3.4 adds a participant lock for MCP servers, a skill filter for A2A executors, and digest-bound approval decisions, alongside several breaking changes to profile requirements, generated `/mcp` endpoints, and resource id validation.

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

01 **Participant lock for MCP servers** NEW 85

MCP servers can be locked to a single participant by passing `participant` to `makeChapMcpServer` (TypeScript) or `make_chap_mcp_server` (Python); any call that names a different `from` is refused, preventing an agent from approving its own work in a human's name.

Ensure an MCP server always acts as a specific agent and cannot approve its own work in a human's name.

javascript

```
import { makeChapMcpServer } from '@brightbeamai/chap-
coordinator';

const server = makeChapMcpServer({
  participant: 'agent:triage',
  // ...other options
});
```

— Names exact functions/params and shows runnable code. v0.3.4

02 **Skill filter for A2A executors** NEW 85

A2A executors can restrict which skills they accept using `skillFilter` (TypeScript) or `skill_filter` (Python); any skill excluded by the Agent Card's filter is rejected with JSON-RPC error `-32601`.

Restrict a Python A2A executor to only the skills declared in the Agent Card's filter, rejecting undeclared skills with -32601.

```python
from chap_coordinator import make_chap_mcp_server

server = make_chap_mcp_server(
    participant='agent:triage',
    skill_filter=['summarise', 'triage'],
)
```

— Exact param names and error code with code example. v0.3.4

03 **Artefact digest binding on approval decisions** NEW

80

Review decisions can be tied to the exact artefact they were made on by sending `approved_artefact_digest` with the `decide.approve` call; if a completed artefact is subsequently revised, a new review round now opens automatically rather than silently inheriting the prior approval.

Tie an approval decision to the exact artefact version it applies to, so that revising the artefact cannot silently inherit the old approval.

```json
{
  "jsonrpc": "2.0",
  "method": "decide.approve",
  "params": {
    "from": "human:alice@example.org",
    "task_id": "task-123",
    "approved_artefact_digest": "sha256:ab12cd34..."
  },
  "id": 1
}
```

— Names the field and method, shows the JSON-RPC call. v0.3.4

04 **Hardened `/mcp` endpoint and startup checks in generated projects**

70

BREAKING

A generated project's `/mcp` endpoint now makes every call as the agent and serves only the tools listed in `AGENT_MCP_TOOLS`; projects generated before 0.3.4 must manually copy the new `/mcp` call and `AGENT_MCP_TOOLS` from the template. A generated server also now refuses to start if `mcp` is enabled and no `agent.uri` is set.

— Names the config key and endpoint but migration steps are manual/vague. v0.3.4

05 **Stricter id validation for whisper, deliberation and handoff resources**

65

BREAKING

An id that is already in use, is non-string, or equals `__proto__`, `constructor`, or `prototype` is now refused with JSON-RPC error `-32602` for whisper, deliberation, and handoff resources.

— Specific error code and rejected id patterns but no action required by users. v0.3.4

THINNER COVERAGE BELOW

06 **Breaking changes to review, whisper, deliberation and handoff profiles**

50

BREAKING

The `review`, `whisper`, `deliberation`, and `handoff` profiles now have changed requirements, and existing coordinators and generated projects serving `/mcp` must be upgraded to 0.3.4.

— States an upgrade requirement but not what the changed requirements are. v0.3.4

└──▷ **BREAKING ON UPGRADE**

! The `review`, `whisper`, `deliberation`, and `handoff` profiles now have changed requirements — existing coordinators and generated projects serving `/mcp` must be upgraded to 0.3.4.

! A generated project's `/mcp` endpoint now makes every call as the agent and serves only the tools listed in `AGENT_MCP_TOOLS`; projects generated before 0.3.4 must copy the new `/mcp` call and `AGENT_MCP_TOOLS` from the template manually.

! A generated server now refuses to start with `mcp` enabled and no `agent.uri` set.

! An id already in use, a non-string id, or an id equal to `__proto__`, `constructor`, or `prototype` is now refused with `-32602` for whisper, deliberation, and handoff resources.

WAS THIS USEFUL? 👍 👎

elementor-mcp

Feedback ⓘ v3.21.0 NOTES ↗ RANK

WordPress MCP server with 234 Free tools and 547 total with Pro for building pages, managing content, and operating your site. Availability depends on installed integrations and enabled tools.

> elementor-mcp's v3.21.0 release adds incident-response capabilities for compromised WordPress sites — compromise detection, one-click hardening, binary file writing — alongside two breaking changes tightening backup and plugin-protection behavior.

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

01 **Compromise detection in scan-security** NEW 95

Adding `checks: ["compromise"]` to a `scan-security` call detects active compromises: backdoor folders with nested `.htaccess`, recently written PHP files with backdated timestamps, cloaked homepages serving different content to Googlebot, publicly downloadable backup archives, suspicious must-use plugins, and re-infection scheduled tasks. Three new malware rules were also added covering split superglobal names, a known injector marker, and includes from compressed streams.

— Exact check parameter and detection mechanisms named v3.21.0

02 **AI-driven site hardening with harden-site** NEW 95

The `harden-site` tool disables XML-RPC, hides the WordPress version from page source and asset links, blocks user enumeration via the REST API and `?author=N`, and hides PHP errors from visitors; each change is reversible from History. The companion `get-hardening-status` tool checks each control as a visitor would see it.

— Named tools, exact controls, and reversal mechanism v3.21.0

03 **Binary file writing via base64 encoding** IMPROVED 85

`write-file` now accepts `encoding: "base64"` to write images, fonts, and ZIP files byte-for-byte. Content is validated strictly, and the existing 5 MB limit applies to the decoded size.

— Exact parameter and limit behavior specified v3.21.0

04 **Auto-enable new tools after updates** NEW 60

A new auto-enable new tools switch on the Tools screen (off by default) lets managed client sites automatically receive new write, database, plugin, and user tools after updates without a manual visit; tools previously turned off stay off.

— UI location given but no API/flag name v3.21.0

THINNER COVERAGE BELOW

05 **wp-config.php excluded from full backups** BREAKING 50

Full backups no longer include the root `wp-config.php` file, which holds the database password and auth keys; `wp-config-sample.php` and same-named files inside plugins are still archived.

— Clear scope but no migration step given v3.21.0

06 **MCP Adapter plugin protected from AI deactivation** BREAKING 25

AI tools can no longer deactivate or delete the MCP Adapter plugin.

— Single sentence, no mechanism or workaround v3.21.0

└──▷ **BREAKING ON UPGRADE**

! Full backups no longer include the root `wp-config.php` file (which holds the database password and auth keys); `wp-config-sample.php` and same-named files inside plugins are still archived.

! AI tools can no longer deactivate or delete the MCP Adapter plugin.

WAS THIS USEFUL? 👍 👎

boost

Feedback ⓘ v2.10.4 NOTES ↗ RANK

Laravel-focused MCP server for augmenting your AI powered local development experience.

> boost's latest release adds a rules re-indexing command and PHPUnit TestCase support in the testing skill, while removing the `application-info` MCP tool and fixing permissions on remote skill scripts.

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

01 **Removal of application-info MCP tool** BREAKING 45

The `application-info` MCP tool has been removed from boost.

– Names the removed tool but no migration path given v2.10.4

02 **PHPUnit TestCase support in testing skill** IMPROVED 40

Unit tests can now extend PHPUnit's `TestCase` within the testing skill.

– Names the class but no further mechanism v2.10.4

03 **Executable permissions preserved on remote skill scripts** IMPROVED 30

boost now preserves executable permissions on remote skill scripts, fixing an issue where such permissions could be lost.

– Describes the fix but no mechanism or scope detail v2.10.4

04 **Command to re-index rules** NEW 20

Adds a command to re-index rules, though the exact command name is not specified in the release notes.

– No command name or usage given v2.10.4

└──▷ **BREAKING ON UPGRADE**

! The `application-info` MCP tool has been removed.

WAS THIS USEFUL? 👍 👎

kubernetes-mcp-server

Feedback ⓘ v0.1.1 NOTES ↗ RANK

Model Context Protocol (MCP) server for Kubernetes and OpenShift

> kubernetes-mcp-server added MCP Apps support with a custom authoring API, RFC 8693 cross-realm token exchange, and new KubeVirt and NetObserv integrations, alongside Kiali filtering/multicluster improvements, structured Tekton PipelineRun diagnosis, stdin-based TOML config, pagination for list calls, and two breaking changes affecting HTTP-mode tooling and parameter shapes.

└──▷ **WHAT SHIPPED · 12 FEATURES** most completely described first ? what's the number?

01 TOML configuration from stdin NEW

75

The server can read TOML configuration from stdin by passing `--config -` on the command line instead of requiring a file on disk, useful for ephemeral or secrets-manager-backed deployments.

Pipe a TOML config into the server at startup instead of pointing to a file on disk — useful in ephemeral or secret-managed environments.

```
$ cat config.toml | kubernetes-mcp-server --config -
```

Pipe a TOML config from a secrets manager or another process without writing it to disk.

```
$ vault kv get -field=config secret/mcp-config | kubernetes-mcp-server --config -
```

— Names exact flag and gives runnable pipe examples. v0.1.1 v0.1.0

02 Cross-realm token exchange (RFC 8693) NEW

75

Adds a `token_exchange.token_url` config key to enable cross-realm token exchange following RFC 8693, letting the server obtain tokens from a different realm or authorization server.

Configure cross-realm token exchange so the server can obtain tokens from a different realm or authorization server.

```yaml
token_exchange:
  token_url:
https://auth.example.com/realms/other/protocol/openid-
connect/token
```

— Named config key, RFC reference, and working config example. v0.0.68

03 **`configuration_view` tool disabled in HTTP mode** BREAKING

60

The `configuration_view` tool is no longer available when the server runs in HTTP mode; any client or workflow relying on it in that mode will stop working after upgrade.

— Names exact tool and mode affected with clear before/after. v0.0.68

THINNER COVERAGE BELOW

04 **NetObserv MCP contract and observability tooling** NEW

55

Adds a NetObserv MCP contract with bounded RBAC metadata declared on OpenShift, and imports observability MCP tooling from `obs-mcp`.

— Names contract source and platform but no usage path. v0.0.68

05 **MCP Apps support with custom authoring API** NEW 50

Adds MCP Apps support, including a custom MCP App authoring API and shared viewers for core Kubernetes tools.

— Names the authoring API and viewers but no example usage. v0.0.68

06 **Kiali filtering and multicluster evaluation** IMPROVED

45

Adds API-based filtering for Kiali to reduce payloads returned to the model by narrowing service-mesh data at the source rather than post-fetch, adds Kiali multicluster evaluations, and filters Kiali tools based on configured URL reachability.

— Explains mechanism but no concrete flags or commands given. v0.0.68 v0.1.1 v0.1.0

07 **Bearer token access on ClusterInspector** NEW 45

Adds `BearerToken` access on `ClusterInspector` so integrations can retrieve bearer tokens programmatically.

— Names the API field but no usage example provided. v0.0.68

08 **Structured PipelineRun diagnosis for Tekton** NEW 40

Adds structured PipelineRun diagnosis for Tekton, giving the server insight into pipeline execution failures and surfacing failure details.

— Describes purpose but no tool name or invocation shown. v0.1.1 v0.1.0

09 **KubeVirt VM console screenshot tool** NEW 40

Adds a VM console screenshot tool for KubeVirt to capture the current state of a virtual machine's display.

— Names the integration and tool but no invocation detail. v0.0.68

10 **Consolidated tool, prompt and resource parameters** BREAKING 35

Tool, prompt, and resource parameters have been consolidated; existing callers using the old parameter shapes may need to update their requests.

— Flags a migration need but no specifics on the new shapes. v0.0.68

11 **Pagination for list-style calls** IMPROVED 30

Adds pagination to list-style calls so large clusters no longer overwhelm the server response.

— States the change but no parameter names or limits given. v0.0.68

12 **In-flight MCP request metric** NEW 20

Tracks the number of in-flight MCP requests as a new observability metric.

— Bare mention with no mechanism or access path. v0.1.1

└──▷ **BREAKING ON UPGRADE**

! The `configuration_view` tool is no longer available when the server runs in HTTP mode — any client or workflow that relies on it in that mode will stop working after upgrade.

! Tool, prompt, and resource parameters have been consolidated — existing callers using the old parameter shapes may need to update their requests.

WAS THIS USEFUL? 👍 👎

rails-ai-context

Feedback ⓘ v5.38.2 NOTES ↗ RANK

Stop AI from guessing your Rails app. 35 MCP tools give coding agents ground truth on demand: schema, models, routes, controllers, views, jobs and conventions.

> rails-ai-context's v5.38.2 release adds a per-answer GitHub issue reporting prompt with an opt-out, enforces strict byte budgets on generated context files, refines root-file content heuristics, overhauls how it writes files across different AI tool adapters, and stops generating the now-redundant .cursorrules file.

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

01 **Per-answer GitHub issue reporting prompt** NEW 86

Every answer now ends with one line asking the AI to offer a GitHub issue when rails-ai-context answered wrong, missed something, or could do better; it appears once per answer, surfaces under a `_report` key in JSON answers, and is absent from composite tool parts. Set `config.report_issue_line = false` to opt out, which saves roughly 40 tokens per answer.

Opt out of the per-answer GitHub issue prompt to keep token usage lean in automated or high-volume agent sessions.

ruby

```ruby
config.report_issue_line = false
```

— Exact config key, JSON field, token cost and example given. v5.38.2

02 **File generation conventions across AI tool adapters** IMPROVED 80

`AGENTS.md` now carries the root content written once for all AI tools; `CLAUDE.md` holds only `@AGENTS.md`; `.github/copilot-instructions.md` stays a full byte-for-byte copy of `AGENTS.md`. Rules files declare one scope — always, by paths, or on request — written by each adapter as `paths:`, `globs:` and `alwaysApply`, or `applyTo`. Overview and tools rules are now written only when `generate_root_files = false`, into `.claude/rules/rails-context.md`, `rails-mcp-tools.md`, `.cursor/rules/rails-project.mdc`, `rails-mcp-tools.mdc`, `.github/instructions/rails-context.instructions.md`, and `rails-mcp-tools.instructions.md`.

— Names every file path, config key, and frontmatter field involved. v5.38.2

03 **Byte budgets for generated context files** IMPROVED 70

Context files now enforce byte budgets: the compact root file stays within 12,000 bytes, full mode within 24,000 bytes, rules files within 8,000 bytes, and a standalone tools rule within 12,000 bytes. List lines are capped at 240 characters, with a `...N more` suffix appended when content is cut.

— Concrete byte limits named but no direct user action to invoke. v5.38.2

---

04 **Root file model and test command heuristics** IMPROVED

65

The root context file now names its 10 most connected models, each on one line, pointing to `rails_get_model_details` where the list is cut. Apps where more than a quarter of model specs use the `user_model_spec.rb` naming pattern now get a test command clause pointing at `rails_get_test_info`.

— Names the triggering heuristics and the tools they reference. v5.38.2

---

05 **Cursor rules file no longer generated** BREAKING

60

`.cursorrules` is no longer written; an upgrade removes the block rails-ai-context wrote from any existing `.cursorrules` file and deletes the file if it held only that block.

— Exact migration behavior described, no manual steps needed. v5.38.2

└──▷ **BREAKING ON UPGRADE**

! `.cursorrules` is no longer written; an upgrade removes the block rails-ai-context wrote from any existing `.cursorrules` file and deletes the file if it held only that block.

WAS THIS USEFUL?  

api-mcp-server

Feedback ⓘ v3.22.1 NOTES ↗ RANK

MCP server that exposes Hostinger's platform API for hosting deployment, CDN control, domain management, and WordPress operations.

> api-mcp-server shipped bundled agent skills for headless deployment guidance, an OAuth sign-in flow, and a new operation to query changelog and roadmap entries.

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

01 **Changelog and roadmap entries operation** NEW 55

Adds a `changelog_list-entries` operation to query changelog and roadmap entries — what's new and what's coming.

— Named operation is directly callable, scope clear v3.20.0

02 **OAuth sign-in flow** NEW 35

Adds an OAuth sign-in flow, implemented in `src/core/oauth.js` and `src/core/oauth.ts`.

— Names implementation files but no flow details or usage v3.22.1

WAS THIS USEFUL? 👍 👎

funplay-cocos-mcp

Feedback ⓘ v0.7.0 NOTES ↗ RANK

The Most Advanced MCP Server for Cocos Creator Editor with `execute_javascript`, prompts/resources, screenshots, asset dependency validation, tool profiles, one-click client configuration, and editor automation.

> funplay-cocos-mcp's v0.7.0 release adds calibrated frame-paced mouse drag timing and strengthens project-scoped MCP URL security with context verification and improved one-click setup.

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

01 Frame-paced mouse drag timing NEW

80

The mouse drag tool now supports `timing: "frame-paced"` in its configuration, enabling calibrated drags with unscaled duration and holds, bounded receipts, request-key deduplication, same-tool status/cancel support, and stale/focus/pause checks to prevent timing drift on slower renderers.

Opt into frame-paced timing for a mouse drag so duration and hold intervals stay unscaled and bounded, avoiding timing drift on slower renderers.

```yaml
timing: "frame-paced"
```

— Exact config flag and runnable example given, with mechanism detailed v0.7.0

02 Write guards for project-scoped MCP URLs NEW

60

Project-scoped MCP URLs now carry structural identity evidence; mutating calls made on legacy URLs require a verified `expected_context` pin, preventing accidental misrouting between projects.

— Names the guard field but gives no setup command v0.7.0

THINNER COVERAGE BELOW

03 One-click setup for project-bound URLs IMPROVED

50

One-click setup now generates project-bound URLs; a Skills notice reports old endpoints and offers combined reconfiguration, while TOML updates retain the conservative URL-block editor.

— Workflow described but no exact command or path shown v0.7.0

WAS THIS USEFUL? 👍 👎

semble

Feedback ⓘ v0.7.1 NOTES ↗ RANK

Fast and Accurate Code Search for Agents. Uses 99% fewer tokens than grep+read

> semble's v0.7.1 release adds an opt-in HTTP transport for its MCP server and introduces a new Semble skill for coding agents.

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

01 **Opt-in HTTP transport for MCP server** NEW 40

semble now supports an opt-in HTTP transport for its MCP server, in addition to the existing default transport.

— Names the surface but no flag or config to invoke it. v0.7.1

02 **Semble skill for coding agents** NEW 20

semble adds a Semble skill, extending what coding agents can invoke through the tool.

— Bare mention, no mechanism or usage detail given. v0.7.1

WAS THIS USEFUL? 👍 👎

gopher-mcp

Feedback ⓘ v0.1.19 NOTES ↗ RANK

C++ MCP SDK - build Model Context Protocol (MCP) servers and clients in C++ / CPP. Enterprise-grade security, observability, connectivity.

> gopher-mcp's v0.1.19 release adds a wide set of MCP protocol capabilities: structured tool output with schema fallback, signed-cursor pagination, form/URL elicitation, completion APIs, a tasks extension for long-running work, cancellable requests, caching hints, display metadata, extension negotiation with W3C tracing, and list-change notifications.

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

01 **Configurable pagination across tools, prompts, resources** NEW

78

Pagination for tools, prompts, resources, and resource templates now uses signed cursors bound to the issuing list and server, and adds client support for `resources/templates/list`.

— Names exact endpoint and cursor-binding mechanism. v0.1.19

02 **Caching hints and server identity metadata** NEW 73

Added caching hints with `ttlMs` and `cacheScope` on cacheable results, plus server identity in result metadata for 2026-07-28 callers.

— Names exact config keys `ttlMs` and `cacheScope`. v0.1.19

03 **Structured tool output with schema fallback** NEW 70

Tools can now return structured output via `outputSchema` and `structuredContent`, with automatic text fallback for older clients that do not support the newer protocol revision.

— Names exact schema fields but no usage walkthrough. v0.1.19

04 **Completion APIs for prompts and resource templates** NEW

65

Added client and server completion APIs for prompt and resource-template arguments via `completion/complete`.

— Names exact `completion/complete` endpoint. v0.1.19

05 **Tasks extension for long-running tools** NEW 65

Added a tasks extension enabling long-running tools with caller-owned tasks, polling, input requests, cancellation, expiry, and task subscriptions.

— Lists mechanisms but no concrete flags or endpoints named. v0.1.19

THINNER COVERAGE BELOW

06 **Extension capability negotiation and trace context** NEW

59

Added extension capability negotiation and W3C trace context propagation with request span hooks on the client and server.

— Names W3C trace context but no concrete hook API. v0.1.19

07 **List-change notifications for tools, prompts, resources** NEW

52

Added tool, prompt, and resource list-change notifications — including registration and removal updates — when the capability is enabled.

— Describes behaviour but no named endpoint or flag. v0.1.19

08 **Cancellable client requests with timeouts** NEW 50

Added cancellable client requests with enforced request timeouts, progress callbacks, and server-side cancellation signals for deferred work.

— Explains mechanism but no named config surface. v0.1.19

09 **Display metadata across protocol types** NEW 50

Added titles, icons, tool behaviour hints, annotations, and nested `_meta` across protocol types, with client and server display metadata.

— Names `_meta` field but little mechanism detail. v0.1.19

10 **Form and URL elicitation schemas** NEW 48

Added form and URL elicitation support with typed fields, single-choice, and multi-select schemas for gathering structured input from users.

— Describes schema types but no API names given. v0.1.19

└──▷ ALSO FROM THESE RELEASES

Run the bidirectional Python SDK interoperability suite alongside protocol conformance tests to verify your server against the 2026-07-28 schema fixtures.

```
$ make test-interop
```

WAS THIS USEFUL? 👍 👎

claudexor

Feedback ⓘ v4.8.2 NOTES ↗ RANK

Multi-harness control plane for Claude Code, Codex, Cursor, and OpenCode: quota-aware rotation across multiple Claude/Codex subscriptions, shared thread context, and cross-model review.

> Claudexor shipped two releases in one day that twice migrate the engine store schema (v4.8.1 to schema 5, v4.8.2 to schema 6), add garbage-collection introspection via a new CLI command and API endpoint, opaque speech-key tracking for streamed Claude drafts, broader Antigravity file access, a Windows executable integrity check, and a run of smaller task-group, timeline, and model-catalog improvements.

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

01 Engine store schema migration to version 5

BREAKING

95

Opening an existing engine store migrates it to schema 5 in one transaction; engines from v4.7.0, any v4.7.1 build, and the unpublished v4.8.0 refuse a data root that v4.8.1 has served. Stop every writer and take a complete backup — see Upgrade backup and restore in `docs/ARCHITECTURE.md` — before upgrading, since replacing only the executable is not a rollback.

— Exact versions, doc reference and concrete upgrade steps given v4.8.1

02 Garbage-collection inspection command and endpoint

NEW

85

`claudexor gc --last` and `GET /v2/maintenance/gc` let operators inspect the last garbage-collection run, including what was held and any errors — useful for headless daemons with no terminal access.

After a scheduled retention pass, check what was reclaimed, what was held, and whether any errors occurred.

```
$ claudexor gc --last
```

Poll the last garbage-collection outcome from the control-plane API — useful when the daemon is headless and no terminal is available.

```
$ curl -s http://localhost:<port>/v2/maintenance/gc | jq .
```

— Runnable CLI command and exact API endpoint given v4.8.2

03 **Engine store schema migration to version 6** BREAKING

85

Opening an existing engine store migrates the schema from version 5 to 6, adding a lower-case `command.task_group_id` column and backfilling live and archived rows; the migration is one-way, so engines at 4.8.1 or earlier refuse a data root that 4.8.2 has served. Task-group cursors issued before the upgrade restart with `stale_cursor`.

— Exact column, version compatibility and cursor behavior given v4.8.2

04 **Antigravity readonly runs read outside workspace** IMPROVED

80

Antigravity readonly runs now read files outside their workspace: the adapter injects read_file(*) into the profile's own `settings.json` before each run, leaving all other settings and write-confinement rules (`read_file` deny rules and `toolPermission: "strict"`) untouched.

— Mechanism and config keys named, but no direct user action v4.8.2

05 **Opaque speech.key tagging on Claude text blocks** NEW

65

Claude text blocks now carry an opaque `speech.key` with a `fragment` , `complete` , or `abandoned` operation recorded in the run journal and timeline, so a consumer can replace a streamed draft with its complete text without merging unrelated messages.

— Mechanism and field named but no user-facing command v4.8.2

THINNER COVERAGE BELOW

06 **Windows PE shape validation at startup** BREAKING

50

Claudexor now enforces a minimal PE shape check on Windows, refusing to start any `.exe` or `.com` that lacks it (#506).

— Names the check and affected file types, no deeper mechanism v4.8.1

07 **Task-group recovery and multi-run waits** NEW

25

Claudexor adds task-group recovery and multi-run waits (#498).

— Named but not explained beyond the title v4.8.1

08 **Bounded macOS conversation feed and engine history reads** NEW — 25

Claudexor adds a bounded macOS conversation feed and engine history reads (#504).

— Named but no detail on bound size or usage v4.8.1

09 **Cursor observed-model catalog proof** NEW — 25

Claudexor adds a Cursor observed-model catalog proof (#509).

— Named but mechanism and scope unclear v4.8.1

10 **Document uploads via MCP and CLI** NEW — 25

Claudexor adds MCP and CLI document uploads.

— Names the two surfaces but no further detail v4.8.1

11 **Grok Build support** NEW — 25

Claudexor adds Grok Build support.

— Named integration, no configuration or usage detail v4.8.1

12 **Browser lane choice for Auto runs** NEW — 25

Claudexor adds Browser lane choice for Auto runs.

— Named option, no explanation of mechanism or UI path v4.8.1

13 **Editable next messages** NEW — 20

Claudexor adds editable next messages (#507).

— Bare feature name, no mechanism or UI path v4.8.1

14 **Bounded run timeline** NEW — 20

Claudexor adds a bounded run timeline (#517).

— Bare feature name with no further explanation v4.8.1

└──▷ BREAKING ON UPGRADE

! Opening an existing engine store migrates the schema from version 5 to 6 (adds a lower-case `command.task_group_id` column and backfills live and archived rows); the migration is one-way — engines at 4.8.1 or earlier refuse a data root that 4.8.2 has served.

! Task-group cursors issued before the upgrade restart with `stale_cursor`.

! Opening an existing engine store migrates it to schema 5 in one transaction; engines from v4.7.0, any v4.7.1 build, and the unpublished v4.8.0 refuse a data root that v4.8.1 has served — stop every writer and take a complete backup (see Upgrade backup and restore in `docs/ARCHITECTURE.md`) before upgrading. Replacing only the executable is not a rollback.

WAS THIS USEFUL?  

spring-plugin

Feedback ⓘ 262.35.123 NOTES ↗ RANK

Spring Explyt — free, open-source (Apache-2.0) Spring & Spring Boot plugin that brings Ultimate-grade tooling to IntelliJ IDEA Community Edition (works in Ultimate too): runtime-accurate bean navigation, inspections & quick fixes, endpoints, Spring debugger, HTTP client, MCP tools for AI agents.

> Spring Explyt adds two MCP tools for AI agents — bean-injection lookup and call-chain tracing — expands the Endpoints tool window with built-in Actuator endpoints, URL-based lookup and functional route listing, and brings bean condition evaluation and a large set of new inspections in line with how Spring Boot actually resolves things at runtime. It also ships OpenAPI 3.2 support, JCEF-less installs for IDEs like Android Studio, and raises its minimum supported IntelliJ IDEA version to 2025.1.

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

01 Endpoints tool window: Actuator, URL lookup, functional routes

NEW 85

The Endpoints tool window now lists built-in Actuator endpoints (`health` , `info` , `metrics` , and others) next to project endpoints, with exposure and access as Boot reads them. You can find an endpoint by pasting a URL from a browser, a log, or a curl command into the IDE, and functional routes — `router { }` , `WebMvc.fn` , Java `RouterFunctions` builders, and nested path() and nest() prefixes — are now listed as Spring serves them.

Locate the endpoint that produced a URL seen in a log or curl command, without manually searching the codebase.

📍 Open the Endpoints tool window, paste the URL into the search field, and the plugin resolves it to the matching request mapping.

— Gives exact UI workflow and names the route-builder APIs covered. 262.35.123

02 New inspections and quick fixes for Spring startup failures

NEW 85

Adds inspections with quick fixes for: bean name collisions that fail startup (two `@Bean` methods with the same name on Spring Framework 6.2+, and a `@Bean` method overridden across two registered configurations); a composed `@Bean` name conflicting with its meta-annotation, failing startup on Spring Framework 6; a stereotype annotation on an abstract class that Spring does not register; a `@Controller` or `@RestController` value that looks like a request path, with a fix moving it to

`@RequestMapping`; an Actuator endpoint extended by more than one `@EndpointWebExtension`, which fails startup; Spring Boot 3.4/3.5 `@MockBean` and `@SpyBean` deprecation, with a fix to `@MockitoBean` and `@MockitoSpyBean`; Spring Boot 3 `management.endpoint.httptrace.*` keys, with a fix to `httpexchanges`; and an enum configuration or hint value not matching Spring's recommended spelling.

— Names every check and its fix, but no demo of the quick fix running. 262.35.123

---

03 **MCP tools for bean lookup and call-chain tracing** NEW

80

Adds `explyt_find_spring_bean`, which tells an AI agent which bean Spring injects at a field or constructor parameter, and `explyt_trace_spring_call_chain`, which shows which tests a change breaks and the `@Transactional` and `@Async` advice around each method.

— Names exact MCP tools and their behaviour but gives no invocation example. 262.35.123

---

04 **Bean condition evaluation matches Spring Boot runtime**

IMPROVED

75

Bean conditions now follow Spring Boot evaluation: `@ConditionalOnProperty` (including repeated usage, `matchIfMissing`, and `${VAR:default}` syntax), `@ConditionalOnBooleanProperty`, composed conditions, and conditions on library classes are all evaluated.

— Names the exact annotations and syntax handled, no worked example. 262.35.123

---

05 **OpenAPI 3.2 support and Swagger preview upgrades** IMPROVED

70

Adds OpenAPI 3.2 support alongside refreshed OpenAPI 3.0 and 3.1 schemas, Swagger UI 5.32.15, and a working OAuth2 Authorize flow in the preview.

— Names versions and the fixed OAuth2 flow but no usage steps. 262.35.123

---

06 **MCP tool contract accuracy and response paging** IMPROVED

65

Endpoint contracts exposed to MCP tools now describe the JSON that Jackson writes, including polymorphic types, Spring Data `Page`, HATEOAS models, naming strategies, and enums. MCP tool answers are paged within a character budget, with a page revision that stays stable while editing unrelated files.

— Explains contract scope and paging mechanism with no tunable surface given. 262.35.123

---

07 **Install without the JCEF module** IMPROVED

60

Spring Explyt now installs in IDEs without the JCEF module (such as Android Studio); the OpenAPI preview, `Run in Swagger` and `Open in UI` gutters, and Spring Initializr wizard remain available only where JCEF is present.

— States the compatibility boundary but offers no workaround. 262.35.123

THINNER COVERAGE BELOW

08 Minimum IntelliJ IDEA version raised to 2025.1 BREAKING

55

This is the last release supporting IntelliJ IDEA 2024.2 and 2024.3; later releases require IntelliJ IDEA 2025.1 or newer.

— Clear version cutoff stated with no migration guidance. 262.35.123

09 Navigate from listeners to published events NEW

45

Navigate from a listener to events that Spring and libraries publish, such as `ApplicationReadyEvent` and `AuthenticationSuccessEvent`.

— Names example events but not the navigation mechanics. 262.35.123

└──▷ **BREAKING ON UPGRADE**

! This is the last release supporting IntelliJ IDEA 2024.2 and 2024.3; later releases require IntelliJ IDEA 2025.1 or newer.

WAS THIS USEFUL? 👍 👎

concord-mcp

Feedback ⓘ v0.18.0 NOTES ↗ RANK

Live messaging for coding agents

> concord-mcp shipped a new messaging v2 system with delivery receipts and private inboxes, Devin CLI project integration, and scriptable cloud CLI commands for status and agent listing.

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

01 Scriptable cloud status and agent listing commands NEW

81

Adds `concord cloud status --json` for machine-readable cloud status output suitable for scripting and automation, and a new `concord cloud agents` subcommand to list cloud agents.

Parse cloud status in CI pipelines or monitoring scripts without scraping human-readable output.

```
$ concord cloud status --json
```

Enumerate all registered cloud agents, for example to audit which agents are connected to a deployment.

```
$ concord cloud agents
```

– Exact runnable CLI commands with clear use cases v0.18.0

THINNER COVERAGE BELOW

02 Messaging v2 with receipts and private inboxes NEW

43

Adds messaging v2 with sent, delivered, and read receipts plus private inboxes for MCP clients.

– Describes what changed but no API or config detail v0.16.0

03 **Devin CLI project integration** NEW 36

Adds a Devin CLI project integration enabling concord-mcp to work with Devin CLI projects, plus Devin CLI global setup and adapter support allowing concord-mcp to configure and interface with Devin CLI at the global level.

— Named integration but no concrete setup steps or commands v0.17.0

NEW INTEGRATION Google Antigravity adapter

NEW INTEGRATION GitHub Copilot CLI integration

v0.15.0 NOTES ↗ STABLE

concord-mcp adds idle-session waking for Copilot CLI via `concord adapters copilot` and for Antigravity via a one-shot inbox monitor, plus curl-based install.

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @concord-ai/concord-mcp@0.15.0
```

git

```
$ git clone --branch v0.15.0 https://github.com/Get-Concord-AI/concord-mcp.git
# already have the repo? check out this version:
$ git checkout v0.15.0
```

signed build provenance · checked 30 Oct Wrong command?

Wake idle GitHub Copilot CLI sessions via `concord adapters copilot` without waiting for a manual restart.

Wake idle Google Antigravity sessions with a one-shot inbox monitor.

Install concord-mcp via curl without needing npm.

WAS THIS USEFUL? 👍 👎

mcp-server Feedback ⓘ v2.4.1 NOTES ↗ RANK

BrowserStack's Official MCP Server

> BrowserStack's MCP server adds accessibility reporting capabilities, letting agents list reports, scans, and builds, and returns significantly more issue detail per call.

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

01 **Accessibility reporting via MCP** NEW 50

The server now exposes listing of web accessibility reports, scans, and builds as MCP capabilities. Accessibility scans also return about 3x more issues per call, with whole-report issue counts surfaced first.

— Describes capability and a concrete improvement ratio, but no named endpoint or flag

v2.4.1

WAS THIS USEFUL? 👍 👎

davinci-resolve-mcp

Feedback ⓘ v4.16.0 NOTES ↗ RANK

davinci-resolve-mcp is an MCP server integration for DaVinci Resolve Studio.

> davinci-resolve-mcp's v4.16.0 release adds stable timeline identification and pinning to call results, a timeline target for DRX grade generation, and small reliability improvements to the duplicate and parameter-handling behavior.

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

01 **Timeline target for DRX grade generation** NEW 90

`drx generate` now accepts `target: "timeline"` to write a grade into `<pTrackVer>`, the block consumed by a timeline `ApplyGradeFromDRX`; the default remains `"clip"`.

Generate a DRX grade into the timeline track slot so `ApplyGradeFromDRX` can consume it directly, instead of writing to a clip.

```
$ {"tool": "drx generate", "params": {"target": "timeline"}}
```

— Runnable example with exact parameter and consumer named v4.16.0

02 **Timeline identification and pinning in call results** NEW 70

Results now include `_operation.timeline` with `{name, id}` identifying which timeline was acted on, and pinned calls add a `pinned`, `restored_current`, or `restore_failed` field so callers can confirm which timeline was actually acted on.

— Fields named but no usage example of a pinned call v4.16.0

03 **make_current option for duplicate command** IMPROVED 70

The `duplicate` command now accepts `make_current` and returns the copy's `id` and the resulting `current` timeline.

— Flag and return fields named but no example call v4.16.0

THINNER COVERAGE BELOW

04 **Ignored parameter warnings in call envelopes** IMPROVED 45

Calls that succeed but carry keys the action never read now return `ignored_params` and an envelope warning instead of silently discarding them.

– Field named but no example triggering it v4.16.0

WAS THIS USEFUL? 👍 👎

octocode

Feedback ⓘ 0.29.0 NOTES ↗ RANK

Structural code intelligence for AI agents — semantic search, knowledge graphs, and a built-in MCP server in one Rust binary. Give Claude, Cursor, and any MCP client a deep understanding of your codebase.

> Octocode shipped three independent capabilities this window: remote LanceDB connectivity, plain version file support in the release command, and configurable Co-authored-by trailers for commits.

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

01 **Plain version file support in release command** IMPROVED

45

The `release` command now reads plain version files in addition to previously supported version formats.

– Names command but not the file format or path 0.28.0

02 **Configurable Co-authored-by trailers in commit command** NEW

45

The `commit` command now supports configurable Co-authored-by trailers, with generic names and email addresses used in examples.

– Names command and feature but not the config mechanism 0.27.0

03 **Remote LanceDB vector store backend** NEW

40

Octocode now supports remote LanceDB as a vector store backend, enabling connections to hosted LanceDB instances instead of only local storage.

– Names the backend but no config key or connection detail 0.29.0

WAS THIS USEFUL? 👍 👎

premiere-pro-mcp

Feedback ⓘ v1.26.0 NOTES ↗ RANK

Adobe Premiere Pro MCP — independent, local-first server for supported workflows. Connect Claude, Codex, or Cursor through a local CEP bridge.

> premiere-pro-mcp's biggest window addition is idempotency-safe retries and background Media Encoder export jobs, alongside new tools for caption retiming, project-structure auditing, CEP panel/transition management, and freeze-frame creation; it also tightened title and effect-keyframe controls and shipped localized onboarding guides in 14 languages.

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

01 Idempotency keys for state-changing tool calls NEW

95

Any state-changing tool call can take an optional `idempotency_key` (1–128 chars of `A-Z a-z 0-9 . _ : -`) so retrying after a timeout never applies an edit twice; replayed responses carry `idempotency.replayed: true`, concurrent retries return `IDEMPOTENCY_IN_PROGRESS`, and reuse with different arguments fails with `IDEMPOTENCY_KEY_REUSED`. Keys are scoped per tool and per MCP session, kept in memory only (200 keys, 1-hour idle expiry, LRU eviction), and stored as SHA-256 hashes.

— Full mechanism, limits, error codes, and field name given v1.25.0

02 Background sequence export jobs NEW

95

Long sequence exports can now run as background Adobe Media Encoder jobs with `export_sequence_job_start`, polled via read-only `get_export_job_status`, which reports `queued`, `encoding` (with progress), `finalizing`, `completed`, `failed`, `cancelled`, or `unknown` without holding an MCP call open for minutes.

— Exact commands and full status enumeration given v1.25.0

03 Project structure audit tool NEW

95

`audit_project_structure` audits the Premiere Project panel against explicit layout rules, checking allowed root names (exact or `*` / `?` globs), extension-to-bin placement, sequences bin, maximum bin depth, and empty bins; it runs read-only or applies up to 200 proposed moves by passing `apply: true` with the returned `planToken`.

— Names tool, rule set, limit, and exact apply mechanism v1.25.0

04 Caption file retiming tool NEW

90

The new `retime_caption_file` tool retimes local SRT or VTT caption files after ripple edits, supporting `shift`, `ripple_remove_ranges`, `ripple_insert`, and

`edit_cue` operations; an optional `fps` param snaps output times to frames, and files are read/written only inside an approved workspace.

— Names tool, formats, operations, and workspace constraint v1.25.0

05 **Bridge telemetry blocking diagnosis** NEW 85

`get_bridge_telemetry` now detects whether a blocking Premiere dialog has frozen the CEP scripting engine, returning a `blocking` diagnosis with `blocking.state` of `idle`, `busy`, `likely_blocked_by_dialog`, `stuck`, `connector_not_running`, or `unknown`, plus an optional `expected_command_ms` threshold, without sending any command to Premiere.

— Names command, states, and threshold param exactly v1.25.0

06 **CEP panel listing and opening** NEW 75

The new `list_extensions` and `open_extension` commands let users list installed CEP panels and open one by its exact ID; opening reports `committed_unverified` because CEP cannot read panel visibility.

— Names exact commands and status code, but no example shown v1.26.0

07 **Unit-aware effect keyframe controls** IMPROVED 75

Effect keyframe tools now accept host-validated numeric pairs with per-component readback, and unsupported color-key writes are refused before mutation; Motion Position and Anchor Point tools accept explicit pixel or normalized units, with omitted units retaining pixel behavior and warning about fractional-looking values.

— Covers mechanism and named tools but no runnable example v1.26.0

08 **Hold-last-value trim policy** NEW 75

Trimming clips with `keyframe_policy: "hold_last_value"` on `trim_clip` automatically samples animated parameters at the edge and inserts hold keys, keeping the revealed range flat and reporting `committed_unverified` when a key readback fails.

— Names exact flag value and failure reporting behavior v1.25.0

09 **Source freeze frame creation** NEW 75

`create_source_freeze_frame` decodes a freeze frame from a project item at a source-media time using local ffmpeg; it can optionally place the result on a timeline track with `place` and add Motion Scale/Position keyframes at clip edges with `zoom`.

— Names tool and its optional placement/zoom parameters v1.25.0

10 **Remove a single CEP transition** NEW 70

The new `remove_transition` command removes a single CEP transition by exact index or an unambiguous contained cut, then verifies the complete remaining span list.

– Names command and verification behavior precisely v1.26.0

11 **Title command gains font, color, position control** IMPROVED

70

The `add_title` command now accepts an optional PostScript font name, size, fill color, and normalized whole-graphic position, with per-field readback for compatible AE-authored templates.

– Lists exact new parameters and scope limitation v1.26.0

THINNER COVERAGE BELOW

12 **Audio placement, loudness, and EDL output controls** IMPROVED

40

premiere-pro-mcp adds safer audio placement, loudness retry, and bounded EDL output controls.

– Names three areas but no mechanism, flags or limits given v1.24.2

13 **CEP panel reinstallation required** BREAKING

30

Connector changes in this release require reinstalling the Premiere and After Effects CEP panels.

– States only the migration requirement, no mechanism v1.26.0

14 **Project-item usage inspection and mutation recovery** NEW

30

premiere-pro-mcp adds project-item usage inspection and recovery of recent mutation outcomes.

– Describes capability in prose only, no named surface v1.24.2

15 **Localized onboarding guides** NEW

25

premiere-pro-mcp now supports localized onboarding guides for 14 major languages.

– Only a bare count is given, no detail on content or access v1.24.2

└──▷ **BREAKING ON UPGRADE**

! Connector changes require reinstalling the Premiere and After Effects CEP panels.

WAS THIS USEFUL? 👍 👎

kinocut

Feedback ⓘ v1.16.5 NOTES ↗ RANK

Guardrailed video editing MCP server for AI agents. FFmpeg, Hyperframes, repurposing tools, Python client, and CLI.

> Kinocut's latest releases focus on precision and safety: an exact frame-accurate probe mode joins a curated default MCP tool set, while a wide set of new environment variables lock vision-model calls, renderer telemetry and the legacy product adapter behind explicit opt-in consent.

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

01 Exact probe mode for video info NEW

88

Enable exact probe mode with `kino info <file> --exact`, Client.info(path, exact=True), or MCP video_info(input_path, exact=true) to report `frame_count`, `frame_count_source: "decoded"`, and the raw `r_frame_rate` / `avg_frame_rate` strings exactly as ffprobe returns them. Unmeasurable values (such as a `0/1` rate) are returned as `null` rather than a default.

Get a precise decoded frame count and exact frame-rate strings for a video file where container metadata is unreliable.

```
$ kino info footage.mp4 --exact
```

— Names exact fields, null semantics, and three equivalent invocation surfaces v1.16.5

02 Privacy-first telemetry and vision-call opt-in controls

88

BREAKING

Vision-model calls for product descriptions and semantic QC are now off by default via `KINOCUT_VISION_MODE=off`, so model calls stay opt-in even when credentials are present in the environment. Revideo installation and rendering telemetry is disabled by default, re-enabled with `KINOCUT_REVIDEO_TELEMETRY=true`. Hyperframes telemetry, automatic skill installation, and passive update checks are suppressed by default, re-enabled with `KINOCUT_HYPERFRAMES_TELEMETRY=true`, which is also required for Docker rendering and for preview versions other than 0.8.104 and 0.8.140. Full configuration reference and network-access limits are documented in `docs/VISION_PRIVACY.md`.

Prevent Kinocut from ever making vision-model calls in a CI/CD pipeline where provider credentials are present but should not be used for automated QC.

```
$ export KINOCUT_VISION_MODE=off
```

Re-enable Revideo renderer telemetry in a development environment where render performance metrics are wanted.

```
$ export KINOCUT_REVIDEO_TELEMETRY=true
```

Allow Docker-based Hyperframes rendering and non-standard preview versions by explicitly opting in to Hyperframes telemetry.

```
$ export KINOCUT_HYPERFRAMES_TELEMETRY=true
```

— Three named env vars, exact defaults, docs path, runnable export commands v1.16.4

03 **Curated default MCP tool registration** NEW 67

Kinocut now registers a curated 20-tool default set for MCP rather than exposing everything. Setting `KINOCUT_TOOLS=all` exposes all 205 MCP tools, enables hidden-tool discovery, and activates the one-call `video_shorts` flow plus the read-only `video_doctor` preflight check.

— Names the env var and tool counts but no usage example given v1.16.5

04 **Explicit consent required for legacy product adapter** BREAKING 67

The optional legacy product adapter now requires explicit local or remote consent before it can run; `use_ai=true`, `require_vlm=true`, and inherited credentials no longer authorize model calls on their own. Local requests are restricted to a loopback Anthropic-compatible endpoint, ignore environment proxies, reject redirects, and never fall back to cloud; a configured local or custom product route can no longer activate the separate fixed-cloud QC worker.

— Clear mechanism and before/after but no config key or command shown v1.16.4

THINNER COVERAGE BELOW

05 **mcp-video compatibility package for agent hosts** IMPROVED 40

Install the `mcp-video==1.6.19` compatibility package to pick up this release's runtime for agent hosts including Hermes.

— Names exact package version but no further detail on changes v1.16.4

NEW SUPPORTED PLATFORM Python 3.14 support for MCP request context and progress reporting

└──▷ BREAKING ON UPGRADE

! `use_ai=true`, `require_vlm=true`, and inherited credentials no longer authorize Kinocut model calls — an explicit mode must be configured for any adapter that is intended to call a model.

! `KINOCUT_VISION_MODE` now defaults to `off`, so any setup that previously relied on vision features activating automatically when credentials were present in the environment will stop calling vision models after upgrade.

! A configured local or custom product route can no longer activate the separate fixed-cloud QC worker.

v1.16.3 NOTES ↗ STABLE

Kinocut adds explicit overwrite consent (`overwrite=true`), protected path enforcement, durable-job ownership checks, and symlink-aware project creation.

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install kinocut==1.16.3
# already installed? upgrade to this version:
$ pipx install --force kinocut==1.16.3
```

npm

```
$ npm install -g kinocut@1.16.3
```

git

```
$ git clone --branch v1.16.3
https://github.com/KyaniteLabs/kinocut.git
# already have the repo? check out this version:
$ git checkout v1.16.3
```

PyPI attestation · signed build provenance · checked 30 Oct Wrong command?

Require `overwrite=true` explicitly before replacing an existing media output, preventing accidental overwrites.

Refuse writes into agent configuration homes and require same durable-job ownership for existing non-media artifacts.

Existing receipts without ownership evidence now fail closed rather than silently

succeeding.

Project creation now checks the completed destination path, including symlink redirects.

└──▷ **BREAKING ON UPGRADE**

! Replacing an existing media output now requires `overwrite=true` explicitly; calls that previously succeeded without it will fail.

WAS THIS USEFUL? 👍 👎

embody

Feedback ⓘ v6.2.76 NOTES ↗ RANK

The most comprehensive MCP toolkit for TouchDesigner

> Embody shipped a fix to keep its own COMP out of externalized TouchDesigner projects (with a breaking cleanup step for over-tagged projects) and added remote project saving over Convoy without requiring an Allow Execute TD Python grant.

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

01 Remote save_project via Convoy NEW

80

A remote node's TouchDesigner project can now be saved without an Allow Execute TD Python grant using convoy_call(operation="save_project"). The release also guards against saves being triggered during Perform Mode.

Trigger a save on a remote Convoy node from an AI session without needing Allow Execute TD Python on the target machine.

```python
convoy_call(operation="save_project")
```

— Names exact runnable call and the permission it bypasses v6.2.75

02 Externalize Full Project excludes Embody's own COMP BREAKING

70

Externalize Full Project, the tagger, and hand-typed tags now skip Embody, `/perform`, `/ui`, `/sys`, and the root, so the `TDXN` COMP holding Embody keeps loose operators and wiring without Embody inside. Projects saved before their first externalization now reopen with tracking intact and rebuild COMPs created after that save. Breaking: projects over-tagged by v6.2.75 or earlier will have TouchDesigner's own COMPs untracked, a TOX container switched to TDXN, and will prompt before deleting copies of Embody's own code — this requires manual confirmation after upgrading.

— Explains mechanism and breaking impact but no runnable command or UI path v6.2.76

└──▷ **BREAKING ON UPGRADE**

! Projects over-tagged by v6.2.75 or earlier will have TouchDesigner's own COMPs untracked, a TOX container switched to TDXN, and will prompt before deleting copies of Embody's own code — requires manual confirmation after upgrading.

WAS THIS USEFUL?  

slack-skills-plugin

Feedback ⓘ v1.5.0 NOTES ↗ RANK

Claude Code, Codex, Cursor, and npx skills plugin that enables your AI tools with a Slack MCP Server and Slack Developer Skills.

> slack-skills-plugin v1.5.0 adds a dedicated skill for deploying Socket Mode apps to Railway or Heroku and a one-line prompt for bootstrapping a Slack agent app in any compatible coding agent.

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

01 deploy-slack-app skill for Railway and Heroku NEW

88

The new `slack:deploy-slack-app` skill deploys a Socket Mode Bolt app to Railway or Heroku as a separate, always-on Slack app with its own app ID, invoking the Slack CLI `deploy` hook without touching the local development app. It supports macOS and Linux.

Install the deploy skill for a specific agent and let it push your Socket Mode app to Railway without maintaining a separate always-on process locally.

```
$ npx skills add slackapi/slack-skills-plugin -y -s deploy-slack-app -a gemini-cli
```

— Names exact skill, CLI hook, platforms and a runnable command v1.5.0

02 One-line agent setup prompt NEW

67

A one-line setup prompt fetched from `https://docs.slack.dev/agent-setup/prompt.md` lets any coding agent session bootstrap a Slack agent app; compatible with Claude Code, Cursor, Codex, and other agents that support `npx skills`.

Paste this into Claude Code, Cursor, or Codex to have the agent install the plugin and scaffold a starter Slack app in one shot.

```
$ Fetch and follow the instructions at https://docs.slack.dev/agent-setup/prompt.md
```

— Names exact URL and compatible agents, with a runnable prompt v1.5.0

WAS THIS USEFUL?  

◆ AI Agent Frameworks

Arcade

Feedback ⓘ updated · Oct 30, 2026 NOTES ↗ RANK

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

> Arcade's biggest change this window is a set of new deployment-level governance controls — egress restriction, worker secret isolation, and cloud workload identity for the engine pod — alongside in-place project API key rotation and credential redaction in logs and tool responses; the release also carries a long list of breaking changes, including toolkit major-version bumps, removed tools, and new Helm upgrade requirements.

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

01 **New deployment configuration options** NEW 95

Several new Helm values configure deployed servers: `coordinator.email` sends invitation emails from deployments; `deployments.network.egress.allow` restricts a deployed server to a specific in-cluster destination without opening broader private-network access; `engine.workerSecretSharing` withholds organization and platform secrets from tool servers registered or deployed through the API; `engine.serviceAccount.annotations`, plus pod labels/annotations and extra volumes, attach cloud workload identity (e.g. AWS IRSA) or a mounted key to the engine pod; `deployments.sizing.*` sets default resource requests and limits for deployed workers, with `deployments.resourceQuota.enabled` now the single quota switch; and environment variable names starting with `ARCADE_RUNNER_` are reserved for the deployment runner so they cannot be set through a deployed server's configuration.

Prevent a high-privilege deployed server from reading organization-wide secrets it does not need.

```yaml
engine:
  workerSecretSharing: false
```

Allow a deployed server to call an internal service (e.g. a private artifact store) without opening broad private-network access.

```yaml
deployments:
  network:
    egress:
      allow: "internal-artifact-
store.infra.svc.cluster.local:443"
```

Attach a Kubernetes service-account annotation so the engine pod uses cloud workload identity (e.g. AWS IRSA) instead of a mounted static key.

```yaml
engine:
  serviceAccount:
    annotations:
      eks.amazonaws.com/role-arn:
arn:aws:iam::123456789012:role/arcade-engine-role
```

— Every config key named verbatim with exact behavior and runnable YAML examples.

snapshot-20261030

02 **Helm upgrade behavior changes** BREAKING 70

Setting `deployments.enabled: true` now installs quota permissions and readiness assumptions that differ from previous behavior and should be reviewed before upgrading; `values.yaml` is now schema-validated, so custom values must be reviewed and corrected before upgrading; and two additive database migrations now run automatically on `helm upgrade` unless `engine.database.migrations` is set to `false`.

— Three named config keys with consequences, but no migration steps given.

snapshot-20261030

03 **Tool removals across integrations** BREAKING 60

Google Calendar's `DeleteEvent` and `FindTimeSlotsWhenEveryoneIsFree`, Workday's `ListTimeOffHistory`, Zoom's `ListUpcomingMeetings`, and several Slack tools with no alias have been removed.

— Names every removed tool and integration but no migration guidance. snapshot-20261030

THINNER COVERAGE BELOW

04 **Project API key management** IMPROVED 50

Project API keys can now be rotated, relabeled, have their expiration changed, or be revived in place, without recreating them from scratch.

– Describes capability but no endpoint or CLI command given. snapshot-20261030

05 **Toolkit version fallback for unversioned execute calls**

IMPROVED 50

Unversioned calls to the execute API can now automatically opt into an older toolkit version when the newest version's schema rejects the call's inputs.

– Explains the mechanism but no endpoint name or parameter given. snapshot-20261030

06 **Credential redaction in logs and tool execution history**

IMPROVED 50

Tool responses, errors, and execution history no longer echo back credentials a tool call carried, and logs from deployed workloads no longer contain the secrets those servers were configured with or handed per call.

– Clear scope and before/after but no specific surface or flag named. snapshot-20261030

07 **Apps interactive tool UI on by default** IMPROVED 45

Apps, the interactive tool UI on the gateway, is now enabled by default for servers deployed through Arcade Deploy, pre-built toolkit workers, and tools discovered through recommendation.

– Names three contexts but no UI path or toggle given. snapshot-20261030

08 **Removed `format=oxp` parameter from tools-listing endpoint**

45

BREAKING

The `format=oxp` parameter has been removed from the tools-listing endpoint.

– Names the exact parameter and endpoint but gives no replacement. snapshot-20261030

09 **Additional OIDC scopes for Google and Microsoft sign-in**

45

BREAKING

Google and Microsoft sign-in now request additional OpenID Connect scopes; new connections write a profile identifier, which requires granting updated sign-in scopes.

– Describes the change but not which scopes or where to grant them. snapshot-20261030

10 **`supportImages.busybox` deprecated** DEPRECATED 35

`supportImages.busybox` is deprecated and will be removed in a future minor release.

– Named config key but no replacement or removal timeline given. snapshot-20261030

11 **Toolkit major-version bump with tool renames**

BREAKING

20

> Toolkits move to new breaking major versions and several rename tools; any caller that references tools by name must review the Upgrade notes before upgrading.
>
> — No specific toolkits, versions, or renamed tools are named. `snapshot-20261030`

└──▷ BREAKING ON UPGRADE

- ! Toolkits move to new breaking major versions and several rename tools — any caller that references tools by name must review the Upgrade notes before upgrading.
- ! `deployments.enabled: true` now installs quota permissions and readiness assumptions that differ from previous behavior — review before upgrading.
- ! `values.yaml` is now schema-validated; custom values must be reviewed and corrected before upgrading.
- ! The `format=oxp` parameter is removed from the tools-listing endpoint.
- ! Google Calendar's `DeleteEvent` and `FindTimeSlotsWhenEveryoneIsFree`, Workday's `ListTimeOffHistory`, Zoom's `ListUpcomingMeetings`, and several Slack tools with no alias are removed.
- ! Google/Microsoft sign-in now requests additional OpenID Connect scopes — new connections write a profile identifier, which requires updated sign-in scope grants.
- ! Two additive database migrations run automatically on `helm upgrade` unless `engine.database.migrations` is set to `false`.
- ! `supportImages.busybox` is deprecated and will be removed in a future minor release.

WAS THIS USEFUL? 👍 👎

sympozium

Feedback ⓘ v0.12.2 NOTES ↗ RANK

sympozium is a coordination layer for multi-agent AI systems

> Sympozium's latest releases add resumable gate handling for stalled harness workflows, OpenTelemetry gen_ai.* tracing conventions for agent spans, active memory control for agents, and per-agent model tuning settings.

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

01 **Per-agent model tuning with maxTokens and temperature** NEW

60

Tune model behavior per agent with `maxTokens` and `temperature` settings.

– Named config settings, but no example of usage shown v0.12.1

THINNER COVERAGE BELOW

02 **In-place gate resume for harness workflows** NEW 55

Harness mode now supports in-place gate retry (resume), allowing a stalled gate to restart without tearing down the running harness. This is backed by a new resume gate that controls whether a paused agent workflow can continue.

– Mechanism described but no flag, command or config key given v0.12.2 v0.12.1

03 **OpenTelemetry gen_ai.* conventions on agent spans** NEW

55

Agent spans now emit OpenTelemetry `gen_ai.*` semantic convention attributes, standardizing tracing output for observability tooling.

– Names the convention but gives no setup or query steps v0.12.2

04 **Agent memory update and forget operations** NEW 45

Agents can now update and forget memory entries, giving agent packs active control over their shared memory pool.

– Describes capability but no API or method named v0.12.1

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.218 NOTES ↗ RANK

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

> HyperFrames split its CLI authentication from HeyGen's shared session, requiring a one-time re-login, while also adding signed-in telemetry identity and clearer film-length diagnostics in Producer.

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

01 Separate OAuth client for CLI authentication

BREAKING

80

The CLI now authenticates with its own HyperFrames account via `hyperframes auth login`, keeping its session independent from the HeyGen CLI so one tool's sign-in no longer overwrites the other's. Existing users must run `hyperframes auth login` once after upgrading, since the old shared HeyGen CLI session is no longer used and cloud work now bills to the HyperFrames plan.

Re-authenticate after upgrading so cloud renders bill to your HyperFrames plan instead of the shared HeyGen session.

```
$ hyperframes auth login
```

– Exact command and before/after billing behaviour given. v0.8.218

THINNER COVERAGE BELOW

02 Producer reports element setting film length

IMPROVED

45

HyperFrames Producer now reports which element set the film length, making it easier to trace unexpected duration in multi-element compositions.

– Describes behaviour but no exact UI path or output format. v0.8.216

03 Telemetry identity for signed-in CLI users

IMPROVED

40

HyperFrames telemetry now identifies signed-in CLI users by their HyperFrames user ID and marks signed-in usage as non-anonymous.

– Mechanism named but no config or command to act on. v0.8.216

└──▷ **BREAKING ON UPGRADE**

! Existing users must run `hyperframes auth login` once after upgrading — the old shared HeyGen CLI session is no longer used, and cloud work now bills to the HyperFrames plan.

WAS THIS USEFUL?  

yoagent

Feedback ⓘ v0.26.0 NOTES ↗ RANK

yoagent is a Rust agent loop that streams from 7 LLM protocols and runs tools until done, from a laptop to a Cloudflare Worker.

> Across v0.25.16 and v0.26.0, yoagent adds typed HTTP status codes to MCP error handling (with a breaking non-exhaustive variant), an origin-scoped redirect policy for MCP-over-HTTP transports, opt-in Unix process-group isolation with automatic cleanup for stdio MCP servers, and gone-server detection that withdraws tools when a stdio server stops responding.

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

01 Process-group isolation and cleanup for stdio MCP servers

NEW 95

Opt in to Unix process-group isolation for an MCP stdio server by passing StdioOptions::new().own_process_group(true) to McpClient::connect_stdio_with(command, args, env, opts) or `StdioTransport::with_options`; the whole group is killed on close() or `drop`. With `own_process_group` set, a launcher process that exits now also immediately kills its entire process group, preventing backgrounded child servers from outliving the session.

Spawn an npx-based MCP server in its own process group so that Ctrl+C in the host terminal does not reach it, and the entire group (npx + node children) is killed when the client is dropped.

rust

```rust
let client = McpClient::connect_stdio_with(
    "npx",
    &["-y", "@modelcontextprotocol/server-filesystem", "/tmp"],
    env,
    StdioOptions::new().own_process_group(true),
).await?;
```

— Named config, constructors, and runnable spawn example across two releases. v0.26.0 v0.25.16

02 Typed HTTP status on MCP errors

BREAKING 90

MCP-over-HTTP errors now carry a typed status via the new `McpError::Http { status, message }` variant and `McpError::http_status() -> Option<u16>` method, covering calls, notifications, connect handshakes, and expired-session 404s;

custom transports can construct one with McpError::http(status, message). Because `McpError` is now `#[non_exhaustive]` with this new `Http` variant, any exhaustive `match` on `McpError` lacking a wildcard arm will fail to compile.

Branch on a 401 or 403 to prompt for credentials instead of parsing error text.

```rust
match client.call(request).await {
    Err(McpError::Http { status, .. }) if status == 401 || status
== 403 => {
        eprintln!("Auth required (HTTP {status}) — check your API
key");
    }
    Err(e) => eprintln!("Other error: {e}"),
    Ok(result) => handle(result),
}
```

— Names the variant, methods, scope, and breaking compile impact. v0.26.0

03 Origin-scoped redirect policy for MCP HTTP transport NEW

90

HttpTransport::with_redirects(Redirects::None | SameOrigin | Any) controls MCP-over-HTTP redirect behaviour; the new default `SameOrigin` restricts redirects to the original scheme, host, and port so custom headers such as `x-api-key` set via `with_header` are never forwarded to another origin.

Prevent custom headers such as `x-api-key` from leaking to redirected hosts by keeping the default `SameOrigin` policy, or opt back to unrestricted redirects explicitly.

```rust
let transport = HttpTransport::new(url)
    .with_header("x-api-key", my_key)
    .with_redirects(Redirects::SameOrigin); // default — safe

// To restore the old behaviour explicitly:
let transport = HttpTransport::new(url)
    .with_redirects(Redirects::Any);
```

— Exact method, enum variants, default behaviour and header example. v0.26.0

04 Automatic gone-server detection for stdio MCP clients NEW

70

A failed write to an MCP stdio server now marks it as gone: McpClient::exited() returns true, and an `McpToolSource` withdraws its tools automatically.

— Named methods but no usage example provided. v0.25.16

└──▷ **BREAKING ON UPGRADE**

! `McpError` is now `#[non_exhaustive]` and gains a new `Http` variant — any exhaustive `match` on `McpError` that lacks a wildcard arm will fail to compile.

WAS THIS USEFUL? 👍 👎

OpenClaw

Feedback ⓘ v2026.9.36 NOTES ↗ CODE ↗ RANK

OpenClaw is a personal AI assistant platform that connects messaging channels, tools, and model providers through a gateway.

> OpenClaw adds workspace-scoped filesystem enforcement for Gateway-built tools, restricting them to the workspace unless explicitly approved.

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

01 Workspace-only filesystem enforcement for Gateway tools NEW

75

Setting `tools.fs.workspaceOnly` confines Gateway-built tools to the workspace: Gateway loopback and skill-dispatch tools now require approval for out-of-scope access, and media tools are kept inside the workspace.

Prevent Gateway-built tools from accessing files outside the workspace, requiring explicit approval for any out-of-scope filesystem operations.

yaml

```yaml
tools:
  fs:
    workspaceOnly: true
```

— Names exact config key and behavior, but limited detail on approval mechanism.

v2026.9.36

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ langchain-core==1.7.2 NOTES ↗ RANK

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

> LangChain's langchain-core 1.7.2 release adds functions for attaching and retrieving notification metadata in pipelines.

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

01 **Notification metadata helpers in langchain-core** NEW

55

New `add_notification_metadata` and `get_notification_metadata` functions let developers attach and retrieve notification metadata within LangChain pipelines.

– Names exact functions but not mechanism or use case `langchain-core==1.7.2`

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ harness-cli/v0.2.3 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 shipped agent cancellation, a DecisionModel human-in-the-loop risk classifier, new vended tools for file editing and graph orchestration, and request-wide limits with a breaking change to interrupt persistence across Graph and Swarm multiagent patterns, alongside smaller additions like S3 Vectors search, Model.info inspection, and runtime tool registry removal.

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

01 Request limits and interrupt persistence in Graph and Swarm

75

BREAKING

Graph and Swarm multiagent patterns now support request-wide limits in Python and TypeScript; Python enforces invoke limits and usage accounting across the whole agent tree, and TypeScript persists subagent and swarm interrupts on the parent agent — a breaking change for callers that relied on interrupt state being local to the subagent or swarm.

– Describes before/after and scope but no code sample harness-typescript/v0.3.2 harness-cli/v0.2.3

02 DecisionModel human-in-the-loop risk classifier NEW

73

A new `DecisionModel` primitive (`LLMDecisionModel` in Python) acts as a human-in-the-loop risk classifier, and is now accepted by `HumanInTheLoop` in both Python and TypeScript. System One providers for decisions were also added in Python.

– Named primitives and integration points but no usage example harness-cli/v0.2.3

03 New graph, read, write, edit vended tools NEW

73

Adds `graph` , `read` , `write` , and `edit` vended tools for agents in Python and TypeScript; the new `edit` tool deprecates `file_editor` (Python) and `fileEditor` (TypeScript).

– Names new tools and deprecated predecessors, no example harness-cli/v0.2.3

04 Cancel a running agent invocation NEW

70

Agents can now be cancelled mid-flight via agent.cancel() in TypeScript, useful when a user aborts a request or a timeout fires.

Cancel a long-running agent mid-flight in TypeScript — useful when a user aborts a request or a timeout fires.

```typescript
import { createHarness } from '@strands-agents/harness'

const agent = await createHarness()
const invocation = agent.invoke('Summarize every file in this repo')

// Cancel from another async context, e.g. on an HTTP abort signal
setTimeout(() => agent.cancel(), 5000)

await invocation
```

Cancel a long-running agent task in TypeScript — useful when a user aborts a request mid-flight.

```typescript
import { createHarness } from '@strands-agents/harness'

const agent = await createHarness()
const task = agent.invoke('Scan and summarise every file in this repo')

// Cancel from another async context, e.g. an AbortSignal handler:
agent.cancel()
```

— Named method with two runnable examples `harness-typescript/v0.3.2` `harness-cli/v0.2.3`

05 **S3 Vectors search strategy for S3Storage** NEW SUPPORTED PLATFORM

NEW

60

Python's `S3Storage` gains a new S3 Vectors search strategy, letting you search S3-backed storage using vector search.

— Named class and strategy but no usage example `harness-cli/v0.2.3`

THINNER COVERAGE BELOW

06 **Model capability inspection via Model.info** NEW 55

In TypeScript, `Model.info` lets you inspect model capabilities and context window sizes.

— Named API but no mechanism or example given harness-cli/v0.2.3

07 **Consolidated agent tool streaming logic** BREAKING 35

Shared agent tool streaming logic is consolidated; code that depended on the previous split streaming paths may need to be updated.

— No specifics on what changed in the streaming paths harness-typescript/v0.3.2 harness-cli/v0.2.3

08 **Runtime removal from Python tool registry** NEW 35

A tool can now be removed from the Python tool registry at runtime, via the new tool registry removal feature.

— No API name or example for the removal mechanism harness-typescript/v0.3.2

09 **Structured output attempt limit** NEW 25

The SDK now lets you cap how many times it retries generating structured output, via the new structured output attempt limits feature.

— No API name, config key, or default limit given harness-cli/v0.2.3

10 **Magic Hour integration added** NEW INTEGRATION NEW 25

Adds the Magic Hour integration from the catalog.

— No detail on what the integration enables harness-cli/v0.2.3

NEW INTEGRATION Sprites integration catalog entry

NEW INTEGRATION strands-pixeltable memory store catalog entry

NEW INTEGRATION Asqav observability integration catalog entry

NEW INTEGRATION BotsRated integration catalog entry

NEW INTEGRATION EmpirioLabs AI model provider catalog entry

NEW INTEGRATION membase-strands memory store catalog entry

NEW INTEGRATION mnemoverse memory store catalog entry

└──▷ BREAKING ON UPGRADE

! Subagent and swarm interrupt state is now persisted on the parent agent in TypeScript — callers that relied on interrupts being local to the subagent or swarm will see changed behavior.

! Shared agent tool streaming logic is consolidated — any code that depended on the previous split streaming paths may need to be updated.

! Subagent and swarm interrupts are now persisted on the parent agent in TypeScript (`feat(tools)!`); code that inspects interrupt state on child agents may need to be updated.

! Shared agent tool streaming logic is consolidated (`feat(tools)!`); custom tool streaming integrations in TypeScript may need to be updated.

`harness-python/v0.3.3` NOTES ↗ STABLE

Strands Agents adds structured output attempt limits, agent cancellation via agent.cancel(), a `DecisionModel` / `LLMDecisionModel` risk classification primitive, System One decision providers, a graph tool (Python and TypeScript), read/write/edit vended file tools replacing `file_editor`, `Model.info` for capability/limit introspection, per-agent-tree invoke limits, and tool registry removal in Python.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch harness-python/v0.3.3
https://github.com/strands-agents/harness-sdk.git
# already have the repo? check out this version:
$ git checkout harness-python/v0.3.3
```

Limit how many times an agent retries structured output generation with a new structured output attempt limit setting.

Cancel a running TypeScript agent event loop with agent.cancel().

Remove a tool from the Python tool registry at runtime with the new tool registry removal API.

Retrieve model capabilities and limits in TypeScript via `Model.info`.

Use `DecisionModel` and `LLMDecisionModel` Python primitives to classify risk, and pass a `DecisionModel` as a `HumanInTheLoop` risk classifier in both Python and TypeScript.

+5 more

└──▷ BREAKING ON UPGRADE

! Using `context_manager` with a `RepositorySessionManager` now raises an error instead of silently succeeding.

! Subagent and swarm interrupts are now persisted on the parent agent in TypeScript — existing code that handled interrupt state only on the child may behave differently.

WAS THIS USEFUL?  

Cloudflare Agents

Feedback ⓘ @cloudflare/think@0.23.0

NOTES ↗ RANK

The Cloudflare Agents SDK builds stateful agents on Workers and Durable Objects, with persistent state, scheduling, WebSockets, and MCP server support.

> Cloudflare Agents shipped a durable Pi-harness agent loop, a reworked Agent Skills API (loadSkillCatalog()/createSkillTools() replacing `SkillRegistry`) with live skill reloading in `@cloudflare/think`, OpenTelemetry span bridging, Computer Workspace attachment, MCP client/OAuth metadata support, and a new Scheduler capability.

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

01 Agent Skills API rework with loadSkillCatalog and createSkillTools

93

BREAKING

Load Agent Skills from Computer Workspace files, R2, or inline code using workspace(), r2(), or manifest() loaders passed into the new loadSkillCatalog() function, then expose them as tools via createSkillTools() (from `agents/skills/tools/ai-sdk`); this replaces the removed `SkillRegistry`. parseSkill() now follows the Agent Skills spec and loads skills that break a rule with warnings. runner() has moved from `agents/skills` to `agents/skills/runner`, and the `agents:skills` Vite import and `agents/skills/compile` entry point are removed.

Load Agent Skills from multiple sources and expose them as AI SDK tools in a single catalog.

typescript

```typescript
import { loadSkillCatalog } from 'agents/skills';
import { createSkillTools } from 'agents/skills/tools/ai-sdk';
import { workspace, r2, manifest } from 'agents/skills';

const catalog = await loadSkillCatalog([
  workspace(),
  r2(env.MY_BUCKET),
  manifest(myInlineSkill)
]);

const tools = createSkillTools(catalog);
```

— Full API surface named with runnable example and migration notes agents@0.32.0

02 **Live skill reload and context delivery in @cloudflare/think**

85

BREAKING

`@cloudflare/think` now reloads getSkills() sources every agent turn, so skills written to `/workspace/.agents/skills` via skills.workspace(this.workspace) are picked up without a restart; skill catalog changes are now delivered to the model as a context reminder instead of rebuilding the cached system prompt. This replaces the removed `skillWorkspace` property and the removed `"skill"` context block type, both of which must be updated in any referencing code.

— Names reload mechanism, path, and removed APIs needing migration

`@cloudflare/think@0.23.0`

---

03 **Agent observability features** NEW

78

Agents now emit structured events — covering state changes, schedules, and MCP connections — that can be consumed for observability or automation, and spans from OpenTelemetry-instrumented libraries can be recorded as Workers trace spans by calling experimental_registerOpenTelemetry() from `agents/observability`.

Bridge OpenTelemetry spans from instrumented libraries into Workers trace spans for end-to-end observability.

typescript

```
import { experimental_registerOpenTelemetry } from
'agents/observability';

experimental_registerOpenTelemetry();
```

— Describes scope only, no event names or consumption API Exact import and call shown, runnable from the example

`agents@0.32.0`

---

04 **Durable agent loop via Pi harness** NEW

77

Run a durable agent loop with the Pi harness via .use(), which uses the Lifecycle's job queue so a run continues after the Durable Object is evicted. Pi packages are optional peer dependencies of `agents` at `~1.4.0`; Pi Durable is experimental and its API can change in minor releases, and each `agents` release supports only one Pi minor version, so upgrading `agents` may require a matching Pi version pin.

— Names the method, dependency pinning and eviction behaviour, no example `product docs`

---

05 **MCP client manager and OAuth metadata support** NEW

75

Agents now include a built-in `MCPClientManager` capability for connecting to MCP servers as tools. A new `clientMetadataUrl` option on

`DurableObjectOAuthClientProvider` supplies OAuth Client ID Metadata Documents, alongside updated `@modelcontextprotocol/client` 2.4.0 and `@modelcontextprotocol/sdk` 1.32.2 dependencies.

Supply an OAuth Client ID Metadata Document URL when constructing an MCP OAuth client provider.

```typescript
import { DurableObjectOAuthClientProvider } from 'agents';

const provider = new DurableObjectOAuthClientProvider({
  clientMetadataUrl: 'https://example.com/.well-known/oauth-client'
});
```

— Names class, option and exact dependency versions with example `agents@0.32.0`

THINNER COVERAGE BELOW

06 **System prompt ordering fix for context blocks** IMPROVED

50

An overridden getSystemPrompt() now always renders above context blocks instead of being dropped once any context block exists.

— Clear before/after behaviour but no reproduction steps `@cloudflare/think@0.23.0`

07 **Scheduler capability in the Lifecycle** NEW

40

Agents now include a Scheduler capability for scheduling durable work from within the Lifecycle.

— Names the capability but not its API or config `product docs`

08 **Computer Workspace attachment for agents** NEW

35

Give an agent a durable file system and code execution environment by attaching a workspace from Cloudflare Computer.

— One-line description, no API name or example given `product docs`

09 **Widened TanStack AI peer dependency range** IMPROVED

35

TanStack AI tools now compile against both `@tanstack/ai` 0.x and 1.x after the peer dependency range was widened to include 1.x.

— Names package and versions but no usage guidance `agents@0.32.0`

└──▷ BREAKING ON UPGRADE

- ! Pi Durable is experimental and its API can change in minor releases; each `agents` release supports only one Pi minor version, so upgrading `agents` may require a matching Pi version pin.
- ! `skillWorkspace` is removed; update any code referencing it.
- ! The `"skill"` context block type is removed; update any context blocks that used this extension type.
- ! `agents/skills` replaces `SkillRegistry` with loadSkillCatalog() and createSkillTools() — any code importing `SkillRegistry` will break.
- ! runner() is moved from `agents/skills` to `agents/skills/runner` — imports from the old path will break.
- ! The `agents:skills` Vite import and `agents/skills/compile` entry point are removed — any build referencing them will break.

WAS THIS USEFUL? 👍 👎

open-multi-agent

Feedback ⓘ v1.23.0 NOTES ↗ RANK

Self-hosted TypeScript agent runtime with durable approvals and verifiable run records. Own it, approve it, audit it.

> open-multi-agent added Zod 4 schema support across its agent and judge interfaces while shipping three breaking changes: a stricter zod peer dependency, narrower typing on core schemas, and new hard caps on coordinator plan size.

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

01 **zod peer dependency now required** BREAKING 78

`zod` is now a required peer dependency with the range `^3.25.0 || ^4.6.6`. Projects depending on zod 3.23 or 3.24 must upgrade zod first; package managers that do not install peers automatically require adding `zod` to your own dependencies.

— Exact version range and clear upgrade action given v1.23.0

02 **Stricter typing on taskProfileSchema and gatePolicySchema** 75

BREAKING

`taskProfileSchema` and `gatePolicySchema` are now typed as `ZodSchema<TaskProfile>` and `ZodSchema<GatePolicy>`. Object-only members such as .extend() and `.shape` no longer type-check on them; code calling those methods must define its own schema.

— Names exact types and broken methods, gives remediation direction v1.23.0

03 **Zod 4 schema support across agent interfaces** NEW 71

Tool `inputSchema` and `outputSchema`, `AgentConfig.outputSchema`, judge `verdictSchema`, and SharedMemory.write() schemas now accept Zod 4 classic schemas alongside Zod 3, and zodToJsonSchema() converts both with the same rules.

— Names all affected schema surfaces but no migration example v1.23.0

04 **Coordinator plan size caps** BREAKING 68

Coordinator plans are now capped: a plan with more than 200 tasks, a task with more than 200 `dependsOn` entries, or more than 64 `requires.requiredTools` or `requires.requiredCapabilities` entries fails validation.

— Exact limits named but no migration path given v1.23.0

└──▷ **BREAKING ON UPGRADE**

! `zod` is now a required peer dependency with the range `^3.25.0 || ^4.6.6`. Projects depending on zod 3.23 or 3.24 must upgrade zod first; package managers that do not install peers automatically require adding `zod` to your own dependencies.

! `taskProfileSchema` and `gatePolicySchema` are now typed as `ZodSchema<TaskProfile>` and `ZodSchema<GatePolicy>`. Object-only members such as .extend() and `.shape` no longer type-check on them; code that called those methods must define its own schema.

! Coordinator plans are now capped: a plan with more than 200 tasks, a task with more than 200 `dependsOn` entries, or more than 64 `requires.requiredTools` or `requires.requiredCapabilities` entries fails validation.

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@7.0.157 NOTES ↗ RANK

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

> Vercel AI SDK graduated two previously experimental APIs to stable, `filterActiveTools` and `LanguageModelStreamPart`, keeping the old names as deprecated aliases.

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

01 **Stable `filterActiveTools` and `LanguageModelStreamPart` APIs**

IMPROVED 60

`filterActiveTools` is now stable and can be imported directly, with `experimental_filterActiveTools` kept as a deprecated alias. `LanguageModelStreamPart` is also now stable, with `Experimental_LanguageModelStreamPart` deprecated.

— Names exact APIs and old/new aliases but no behavior change. ai@7.0.157

WAS THIS USEFUL? 👍 👎

LiveKit Agents

Feedback ⓘ livekit-agents@1.8.10 NOTES ↗

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

LiveKit Agents adds bitrate/encoding-mode control to RecorderIO and exposes the Mistral Conversations API store option on the LLM plugin.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch livekit-agents@1.8.10
https://github.com/livekit/agents.git
# already have the repo? check out this version:
$ git checkout livekit-agents@1.8.10
```

Control recording bitrate and encoding mode in `RecorderIO` with new `bit_rate` and `constant_bit_rate` parameters to produce constant-bitrate audio files compatible with Egress's 128 kb/s encoding.

Expose the Mistral Conversations API `store` option on the LLM plugin.

NEW INTEGRATION Reson8 STT plugin

NEW INTEGRATION Mistral Voxtral 3 STT and TTS models

NEW INTEGRATION Inworld TTS model inworld-tts-2-flash

NEW INTEGRATION AssemblyAI typed TTS support via LiveKit Inference

└──▷ BREAKING ON UPGRADE

- ! The Mistral plugin's `api_mode` parameter is now ignored; migrate Chat Completions usage to the Conversations API.
- ! The Mistral STT known model IDs `voxtral-mini-latest` and `voxtral-mini-2507` are removed; explicitly select a batch model when calling recognize().
- ! Mistral realtime streams without VAD or server-side turn detection now require an explicit stream.flush() call.
- ! The Mistral plugin now requires `mistralai>=3.5.0,<4.0.0`; defaults change to Voxtral 3 STT/TTS and `mistral-small-latest`, and Silero is no longer loaded automatically.

WAS THIS USEFUL? 👍 👎

cotal

Feedback ⓘ v0.127.0 NOTES ↗ RANK

The open standard for agent coordination

> cotal shipped a wide range of CLI and API additions this window, headlined by mandatory IdP sign-in confirmation for `cotal manager start` / `restart` (a breaking change requiring `confirmLogin` on custom auth providers), new systemd-based daemon setup/install/uninstall commands for headless machines, and expanded `cotal service-endpoint` caller management (`callers`, `grant`, `revoke`). It also added conversation participant binding, broker payload limits, nats-server log rotation, typed membership feed callbacks, and a Node.js 22.10 minimum version requirement.

└──▷ **WHAT SHIPPED · 29 FEATURES** most completely described first ? what's the number?

01 Caller grant, revoke, and list subcommands in service-endpoint

NEW 95

`cotal service-endpoint` gains caller management subcommands: `cotal service-endpoint callers <proposalId>` lists grantees for an approved service; `cotal service-endpoint grant <proposalId> --to <owner.actor> --lifecycle <uid> --commands <a,b>` grants a specific agent lifecycle access to invoke named commands under an approval's caller ceiling; and `cotal service-endpoint revoke <grantId>` stops new issuance for an existing grant. Duplicate grant requests are detected and suppressed automatically, and retries after a lost response replay the same request safely.

Audit which agents currently have caller access to an approved service before making changes.

```
$ cotal service-endpoint callers <proposalId>
```

Grant a specific agent lifecycle the right to invoke a subset of commands within an approval's caller ceiling.

```
$ cotal service-endpoint grant <proposalId> --to alice.agent --lifecycle <uid> --commands read,write
```

Revoke an existing grant to stop a lifecycle from receiving new call authorisations.

```
$ cotal service-endpoint revoke <grantId>
```

— Three runnable subcommands with full flag syntax and safety mechanism. v0.120.0

02 **cotal daemon setup for headless machines** NEW 95

`cotal daemon setup` connects a headless machine to a space without a local browser: it installs the daemon as a systemd user service, signs in via device code, registers it with the space, and waits for owner approval. Control the approval wait time with `--wait <seconds>` (default 900); pass `--wait 0` to report current progress without blocking, and use `--json` to receive a single structured document on stdout.

Attach a CI runner or headless server to a space from the machine itself, then approve on a separate device using the printed link and code.

```
$ cotal daemon setup --json
```

Cap how long setup waits for the space owner to approve — useful in automated pipelines where you want a quick status check rather than an indefinite block.

```
$ cotal daemon setup --wait 120
```

— Full flag set, defaults, and two runnable examples provided. v0.119.0

03 **Systemd daemon install/uninstall management** NEW 95

`cotal daemon install` sets up the Linux machine daemon as a self-recovering systemd user service — it writes a systemd user unit, enables linger for the invoking user, and starts it; `cotal daemon uninstall` reverses this. Both commands accept `--dry-run`. `@cotal-ai/seat` now exports `flockExclusiveNb`, an exclusive non-blocking flock(2) call, and `@cotal-ai/cli` now exports the systemd user-unit primitives the daemon install uses, giving other callers one shared mechanism for writing and registering units.

Set up the cotal daemon to start automatically on login and recover from crashes without manual intervention on a Linux host.

```
$ cotal daemon install
```

Preview what the daemon install will do without writing any files or enabling linger — safe to run before committing to the change.

```
$ cotal daemon install --dry-run
```

Stop the managed daemon, disable autostart, and clean up the unit — useful when decommissioning a Linux machine or switching to manual daemon management.

```
$ cotal daemon uninstall
```

— Two commands, a flag, and two exported library primitives all named. v0.118.0

04 **IdP sign-in confirmation for manager start/restart** BREAKING 90

`cotal manager start` and `restart` now validate the cached IdP session before saving or launching on a user-auth space — an expired or revoked sign-in returns `AUTH_REQUIRED` (exit 2) and prompts `cotal login --idp <base>` instead of silently failing or leaving the manager running. This requires `AuthProvider` implementations to provide a `confirmLogin` method; providers that do not implement it cause `manager start` and `restart` to refuse with `UNSUPPORTED`.

Recover from a rejected cached sign-in when starting the manager on a user-auth space — the new `AUTH_REQUIRED` (exit 2) tells you exactly what to run next.

```
$ cotal manager start && echo 'manager up' || cotal login --idp
<base>
```

— Exact error codes, remedy command, and breaking interface change all named. v0.122.0

05 **Broker message size limit via --max-payload** NEW 85

`cotal up --max-payload <bytes>` sets the broker's own message size limit (up to 67108864 bytes); cotal records it in `.cotal/broker-policy.json` via `readBrokerPolicy` so a later `cotal up` on the root reapplies the limit without the flag. The minimum for endpoint callers is 2098176 bytes.

Cap broker message size when standing up a space that uses service endpoints — the minimum for endpoint callers is 2098176 bytes.

```
$ cotal up --max-payload 4194304
```

— Exact flag, limits, and config file path all named with example. v0.119.1

06 **Node module compile cache for Claude Code connector seat** NEW

85

Node module compile cache is enabled for the Claude Code connector seat via the new `dist/seat.cjs` entry point, which stores compiled modules in `~/.cotal/compile-cache` (created 0700); disable it with `NODE_DISABLE_COMPILE_CACHE=1` or override the path with `NODE_COMPILE_CACHE`.

Opt out of the new compile cache when profiling the Claude Code connector seat to isolate cold-start behaviour from cached compilation.

```
$ NODE_DISABLE_COMPILE_CACHE=1 node dist/seat.cjs
```

— Entry point, path, permissions, and both env vars named with example. v0.116.0

07 **Timeout remedy guidance for invoke commands** IMPROVED

80

When `cotal invoke` or `cotal endpoint invoke` times out after reading the endpoint contract but before publishing the command, the error now names a longer `--timeout` value as the remedy — e.g. `cotal invoke manager ps --timeout 10000` — and `--json` output carries it as `error.remedy`. Separately, when no instance answers the contract read in time, the command now returns an actionable `TARGET_UNAVAILABLE` error that names the `--timeout` flag and suggests a rerun with a longer value.

Diagnose a publish-deadline timeout in a script by reading the structured remedy from `--json` output instead of parsing human-readable text.

```
$ cotal invoke manager ps --timeout 10000 --json
```

Invoke a contract with an explicit deadline so that a slow or absent instance returns a clear, actionable error instead of a silent nothing serving it response.

```
$ cotal invoke <contract> --timeout 30
```

— Exact error codes, flag, and two runnable example commands given. v0.121.1 v0.117.0

08 **Undecodable key reporting in cotal channel** NEW

80

`cotal channel` now lists registry keys whose values cannot be decoded, showing the decode error, and `readChannelRegistry` returns those keys in an `undecodable` field; `cotal up`, `cotal channel set`, and `channel defaults` now refuse with `CONFLICT` rather than silently overwriting corrupted bytes.

— Named command, field, and error code; no sample output shown. v0.116.0

09 **Rate limiting extended to retire and verify-enrollment endpoints**

IMPROVED 80

`POST /interactive-lifecycle/retire`, `POST /managed-lifecycle/retire`, and `POST /manager-service-authority/verify-enrollment` now count missing or wrong loopback capabilities in the same rate-limited invalid-capability window as `POST /exchange` (first 30 per rolling minute get 401, further requests get 429).

— Exact endpoints and rate-limit thresholds named, no client example. v0.116.0

10 **Fixed service add --preset broker flag mapping** IMPROVED 75

`cotal service add <name> --preset broker` works again — camelCase preset options such as `logFile` now correctly map to their dashed flags (e.g. `--log-file`) instead of refusing the preset as `UNSUPPORTED`.

Add a broker service preset — camelCase options like `logFile` now resolve to their dashed equivalents automatically.

```
$ cotal service add my-broker --preset broker --log-file
/var/log/cotal-broker.log
```

— Runnable command and exact flag mapping shown. v0.121.2

11 **Typed MembershipChange feed callbacks** IMPROVED 75

`watchMembership` now delivers each feed entry to its callback as a typed `MembershipChange` object — containing the decoded member record or its deletion, the heartbeat's `asOf`, and a `replayed` marker — instead of a bare tick. CotalEndpoint.readMembership() now reads the membership feed using paged batched Direct Gets, binding no consumer; the observer and admin profiles gain `$JS.API.DIRECT.GET.KV_cotal_membership_<space>`.

— Named types, fields, and capability string; no example call. v0.116.0

12 **Grant visibility in cotal whoami** NEW 70

Run `cotal whoami` to see the signed-in user's current read, post, direct-message, and capability grant — the stock issuer returns the authoritative grant beside each fresh bearer, while unavailable or older issuers report unknown instead of inferring from the token.

— Runnable command with described issuer behaviour, no sample output shown. v0.121.2

13 **Endpoint kind validation via assertValidKind** IMPROVED

70

`assertValidKind` now validates endpoint kinds (`"agent"` or `"endpoint"`) across `CotalEndpoint` , `configFromEnv` , `loadAgentFile` , and `saveAgentFile` , so `COTAL_KIND` and `card.kind` refuse any other value.

— Names every validated surface but gives no example invocation. v0.121.2

14 **PinnedFetchRefused distinct error for pinned fetch refusals**

NEW

70

A pinned fetch refused by policy, a non-HTTPS/non-loopback target, or a redirect response now throws `PinnedFetchRefused` with a distinct message, letting `meshes add` , `meshes add --from` , and `space add --from` tell policy refusals from transport failures via `instanceof` .

— Named error type and three commands; no example try/catch shown. v0.116.0

15 **Credential-expiry warnings on up --no-manager** NEW

65

`cotal up --no-manager` now warns on authenticated starts and refreshes that no local process renews the daemon credentials, naming their stored expiry times and the same-store manager requirement.

Avoid silent credential expiry in broker-only daemon setups by catching the warning early at daemon start.

```
$ cotal up --no-manager
```

— Runnable command shown but warning content only described, not quoted. v0.120.2

16 **Per-stream color output detection** IMPROVED

65

Color output now follows each stream's own terminal status — `NO_COLOR` , `FORCE_COLOR` , `--no-color` , and `TERM=dumb` apply to both stdout and stderr independently, so redirecting either stream produces plain text.

— Env vars and flag named but no example command shown. v0.119.1

17 **assertValidName expanded character blacklist** IMPROVED

65

`assertValidName` now refuses names containing VT, FF, NEL (U+0085), LINE SEPARATOR (U+2028), and PARAGRAPH SEPARATOR (U+2029) in addition to CR and LF; the `LINE_BREAK` class is now exported from `@cotal-ai/core` .

— Exact disallowed characters and export named, no example invocation. v0.116.0

18 **Single-read roster and channel registry on user-auth mesh commands**

IMPROVED 65

On a user-auth mesh, `cotal send`, `cotal endpoints`, `cotal personas list --running`, and `cotal channels list` now read the roster and channel registry once on the user's bearer and leave no watch or scan consumer behind.

— Four commands named with behaviour change, no before/after numbers. v0.116.0

19 **esbuild minify-whitespace bundling fix** IMPROVED 65

Connector bundles are now built with esbuild's `--minify-whitespace`, keeping all identifiers while dropping comments that caused V8 to store bundles as two-byte strings; text encoding of the 19 built bundles can be verified with `pnpm smoke:connector-bundle-text`.

— Build flag and verification command named, no measured size impact. v0.116.0

20 **Node.js minimum version raised to 22.10** BREAKING 65

The minimum Node.js version required rises from 22 to 22.10 (`engines.node` updated in every package); the `cotal` binary and the `dist/seat.cjs` seat entry now refuse to run on older runtimes, reporting the required floor version with remediation instructions.

— Exact version floor named; no remediation command text quoted. v0.116.0

21 **Terminal interaction hardening** IMPROVED 60

`terminal.tail` is now validated in `orca.readScreen`, so malformed `terminal read --screen --json` responses throw a named error instead of a raw `TypeError`; terminal text input at Claude selection menus is now refused, with guidance to use `cotal attach --name <seat>` and arrow keys instead.

— Named surfaces for both fixes but no example output or command. v0.121.2

THINNER COVERAGE BELOW

22 **Remove-persona Manager command** NEW 55

Remove personas via the typed `remove-persona` Manager command, which carries existing persona authority and handles invalid legacy names.

— Command named but no full syntax or example invocation shown. v0.125.0

23 **Wildcard channel-pattern support in conversation selections**

NEW 55

Conversation selections now accept channel patterns such as `team.*` and `team.>`, granting every matching channel — including channels created after approval — with the

pattern carried as-is in the issued grant.

— Pattern syntax named but no example grant command shown. v0.117.0

24 **Participant binding for hosted workflows** NEW 50

Bind an authorized personal native conversation to a hosted workflow using participant(), receiving workflow turns through a finite, exact-lifecycle lease without gaining manager seat custody.

— No usage example or API signature shown, only a one-line description. v0.127.0

25 **Instance-read and instance-list via public service endpoint** NEW 50

Owner-authorized `instance-read` and cursor-paginated `instance-list` operations, including current approval and registration state, are now accessible through the public service endpoint host.

— Operation names given but no endpoint path or example call. v0.120.2

26 **Native nats-server file logging and rotation** NEW 45

cotal now supports native nats-server file logging and rotation settings, persisted across controller stack restarts, with the broker log and startup diagnostics files kept private.

— No config key, flag, or file path named for enabling it. v0.121.0

27 **Loopback activation for hosted auth services** NEW 35

A capability-gated, account-bound loopback operation now lets separately hosted auth services activate one managed lifecycle through cotal's authority fence.

— No endpoint, flag, or example given — purely prose description. v0.123.0

28 **Permission-request handling for hosted gateways** NEW 35

A hosted gateway now accepts a permission request at connection time via the connector-core, enabling access control decisions at the moment a client connects.

— No API, flag, or example shown — prose description only. v0.118.2

29 **managerCallerBinding exposes credentialId** IMPROVED 30

`managerCallerBinding` now also returns the bearer's `credentialId` when the issuer stamped one.

— One-line internal change with no usage context given. v0.116.0

└──▷ **BREAKING ON UPGRADE**

! `AuthProvider` gains a required `confirmLogin` method; custom auth providers that do not implement it will cause `cotal manager start` and `restart` on user-auth spaces to refuse with `UNSUPPORTED`.

! The minimum Node.js engine version rises from 22 to 22.10 (`engines.node` updated in every package); the `cotal` binary and the `dist/seat.cjs` seat entry refuse any runtime below this floor.

WAS THIS USEFUL? 👍 👎

Agno (formerly Phidata)

Feedback ⓘ v3.1.12 NOTES ↗

RANK

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

> Agno's v3.1.12 release adds native human-in-the-loop approval routing for Claude and Codex agents, support for shipping inactive Routines, headline numbers on charts, and sortable company-routine listings.

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

01 Approval routing for Claude and Codex agents

BREAKING

95

Permission requests from `CodexAgent` and Claude agents can now be routed through the AgentOS approval UI: set `approval_policy='on-request'`, `'untrusted'`, or `'never'` on `CodexAgent`, or use Claude's `permission_mode` with explicit permission callbacks. `hitl_timeout` (default 900s) bounds how long a run waits for approval. A `CodexAgent` created with no approval settings now runs under the native `on-request` policy by default, so a permission request that previously went nowhere will now pause the run instead of continuing.

Give a CodexAgent a human-in-the-loop approval policy so any untrusted action pauses for review in the AgentOS UI rather than proceeding automatically.

python

```python
from agno import CodexAgent

agent = CodexAgent(
    approval_policy="untrusted",
    hitl_timeout=1800,
)
```

— Names exact flags, default values and the default-behavior change v3.1.12

02 Disabled Routines at creation NEW

80

A Routine can be shipped inactive by setting Routine(enabled=False) in Python or YAML bootstrap; it is created atomically, skipped by the scheduler, and can be enabled later from the UI. See `cookbook/05_agent_os/30_company/11_disabled_routine.py` for a worked example.

— Names config key, file path, and scheduler behavior v3.1.12

03 Headline value on chart blocks NEW

75

ChartTools.create_chart() accepts an optional `value` argument (e.g. `'$22.3K'` or `'94%'`) that is emitted as `'value'` in the chart block and rendered beside the chart title in Agno OS.

Display a single summary figure beside a chart title so the viewer can read the key number at a glance.

```python
from agno.tools.chart import ChartTools

chart_tools = ChartTools()
chart_tools.create_chart(
    title="Monthly Revenue",
    data=revenue_data,
    value="$22.3K",
)
```

— Names exact method, argument, and example values v3.1.12

04 **Sortable company routines endpoint** NEW 70

`GET /company/routines` can now sort results by `next_run_at`, `schedule_type`, or `enabled`, with paused and on-demand Routines appearing last regardless of sort direction.

— Names endpoint and exact sortable fields v3.1.12

└──▷ **BREAKING ON UPGRADE**

! A `CodexAgent` with no approval settings now runs under the native `on-request` policy; a permission request that previously went nowhere will now pause the run instead of continuing.

WAS THIS USEFUL? 👍 👎

Feynman

Feedback ⓘ v0.5.60 NOTES ↗ RANK

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

> Feynman v0.5.60 adds three selectable web search providers and a new fallback research model.

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

01 New web search providers: cloudflare, searchapi, tabstack

NEW 90

`feynman search set` now accepts `cloudflare` (requires the `CLOUDFLARE_ACCOUNT_ID` environment variable), `searchapi` (Google results via SearchApi), or `tabstack` (returns a single cited answer); each provider only runs when explicitly selected.

Route web searches through Cloudflare's search API instead of the default provider, scoping it to a specific account.

```
$ export CLOUDFLARE_ACCOUNT_ID=<your-account-id>
feynman search set cloudflare
```

Switch to SearchApi (Google results) when you need broader web coverage than the default provider delivers.

```
$ feynman search set searchapi
```

— Exact command, flag values, and required env var given v0.5.60

THINNER COVERAGE BELOW

02 MiniMax M3.1 Flash research model fallback IMPROVED

35

MiniMax M3.1 Flash is now used as a research model fallback, added after MiniMax M3.

— Names the models but no config or command to invoke it v0.5.60

WAS THIS USEFUL? 👍 👎

pi-fabric

Feedback ⓘ v0.130.0 NOTES ↗ RANK

A programmable tool and agent runtime for Pi

> pi-fabric shipped a v2 managed-host contract adding CPU slice control, provider guidance, a skill reader and foreground tools, alongside new root-session discovery, opt-in action-policy hooks, opt-in durable-worker recovery, and stronger mesh-write durability.

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

01 Managed-host contract v2 configuration surface NEW

90

Managed hosts gain a new set of config keys: `executor.maxCpuSliceMs` (defaults to 1000 ms) caps how long a managed host may hold a CPU time slice; `managedHost.guidance` exposes provider-accurate guidance and can be paired with an optional `skillReader`; and `managedHost.foreground` exposes foreground tools on managed hosts alongside the existing `fabric_exec` tool.

Cap how long a managed host may hold a CPU slice to avoid runaway execution in latency-sensitive environments.

```json
{ "executor": { "maxCpuSliceMs": 250 } }
```

Choose which runtime owns compaction for a managed host — use `pi` when the host should control its own settled lifecycle.

```json
{ "managedHost": { "compaction": "pi" } }
```

— Named config keys with runnable examples and file path. v0.127.0

02 FABRIC_MANAGED_HOST_VERSION bump to 2 BREAKING

70

The managed-host contract version assertion must be updated to `2` — existing assertions pinned to `1` will fail validation after upgrading to v0.127.0.

— Exact env var and required migration step named. v0.127.0

03 Opt-in durable-worker recovery NEW

65

Durable-worker recovery is now opt-in, with host-owned launch authority, safe startup-loss settlement, and correct external-stop handling. Recovery receipts pin the separate

recursive `fabric_exec` permission so revoked authority cannot be restored, and existing unsupported or legacy recovery records fail closed on upgrade.

— Detailed mechanism but no flag or command to enable it. v0.130.0

04 **Opt-in action-policy hooks for extensions** NEW 60

Extension action-policy hooks are now opt-in, with cancellation-safe approval handling and fail-closed malformed decisions. While hooks are active, streaming speculation is disabled so providers cannot run ahead of policy checks.

— Rich behavioral detail but no named flag to enable it. v0.130.0

THINNER COVERAGE BELOW

05 **Mesh write durability and locking** IMPROVED 50

Optional fsync durability for mesh writes now includes async atomic writes and pre-existing directory barriers. The stronger short-lock exclusion guarantee requires upgrading all writers sharing the store; old binaries retain their former race and stale-claim behavior.

— Mechanism and upgrade requirement given, no config name. v0.130.0

06 **Departed root-session discovery** NEW 45

Departed root sessions can now be discovered via `agents.departed`.

— Named API surface but no usage example shown. v0.130.0

07 **Child tool restriction preservation and diagnostics** IMPROVED 45

Child tool restrictions are now preserved through resident dispatch and native wildcard/session selection, with clearer denied-tool diagnostics.

— Describes scope of fix but no reproducible steps. v0.130.0

08 **Bounded progress update handling** IMPROVED 30

Bounded complete progress updates now handle large and escaped phase labels and preserve deadline metadata.

— Brief fix description with no concrete surface named. v0.130.0

└──▷ **BREAKING ON UPGRADE**

! The stronger short-lock exclusion guarantee requires upgrading all writers sharing the store; old binaries retain their former race and stale-claim behavior.

! Existing unsupported or legacy recovery records fail closed on upgrade.

! Update `FABRIC_MANAGED_HOST_VERSION` assertions to `2` — existing assertions pinned to `1` will fail validation after upgrading to v0.127.0.

WAS THIS USEFUL?  

trigger.dev

Feedback ⓘ v4.8.2 NOTES ↗ RANK

Trigger.dev – build and deploy durable AI agents and workflows

> trigger.dev's v4.8.2 release adds Bun 1.4 runtime support, immediate run-requeue controls, a Convex integration, ECDSA P-384 webhook verification, sub-agent response streaming, and new limits alerting, alongside breaking changes to dashboard API token access and the ClickHouse event store.

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

01 Bun 1.4 runtime for tasks NEW

91

Deploy tasks on Bun 1.4.4 in a Bun-only image (no Node.js binary) by setting `runtime: "bun-1.4"` in your task config; `runtime: "bun"` stays on Bun 1.3 with Node.js, and `bun-legacy` is accepted as an explicit alias for that runtime.

Deploy a task on Bun 1.4 (Bun-only image, no Node.js) for faster cold starts and a leaner container.

typescript

```typescript
export const myTask = task({
  id: "my-bun-task",
  runtime: "bun-1.4",
  run: async (payload: { message: string }) => {
    console.log(payload.message);
  },
});
```

— Exact config key, values and runnable task example given v4.8.2

02 Immediate requeue for unstarted runs NEW

86

`SupervisorHttpClient.releaseRun` lets a worker immediately requeue a dequeued run it never started instead of waiting for the pending execution timeout; self-hosted supervisors can enable this automatically via `RELEASE_UNSTARTED_RUNS_ENABLED=true` for runs whose container could not be created and `RELEASE_FAILED_POD_RUNS_ENABLED=true` for pods that failed before starting.

— Named client method and two exact env vars v4.8.2

03 ClickHouse v2 event store required BREAKING

86

The legacy `clickhouse` event store is retired and task events now go to `clickhouse_v2`; self-hosted deployments using

`EVENT_REPOSITORY_DEFAULT_STORE=clickhouse` must switch the value to `clickhouse_v2` and copy older events per the self-hosting guide, or those events will not appear.

— Exact env var, old/new values and migration guidance given v4.8.2

04 **Streamed sub-agent replies in parent tool calls** NEW

80

Stream a sub-agent's reply through a parent tool call with `chat.forwardSubAgent`, then render it incrementally using useSubAgentMessages() from `@trigger.dev/sdk/chat/react` or createSubAgentMessages() outside React.

Stream a sub-agent's reply chunk-by-chunk through a parent agent tool call to avoid re-sending the full message on every token.

```typescript
execute: async ({ topic }, { toolCallId, abortSignal }) => {
  const stream = await researcher.sendMessage(topic, {
abortSignal });
  return await chat.forwardSubAgent(stream, { toolCallId });
},
```

— Named functions with a runnable code example v4.8.2

05 **ECDSA P-384 webhook signature verification** NEW

78

Verify ECDSA P-384 signatures on asymmetric webhook sources with `algorithm: "ecdsa-p384"` (SHA-384); signatures are accepted in DER or raw `r||s` form, and providers that prehash with SHA-256 are supported via `prehash: "sha256"`.

— Exact config values named but no runnable snippet v4.8.2

06 **Limits alerts and queue/webhook metrics** NEW

76

New Limits alerts let you watch limits that can interrupt or degrade runtime flow — individually or combined with `any` or `all` logic — from the dashboard, API, and agent. The `metrics` query table now includes queue size (`queue.size.*`) and webhook limit metrics (`webhook.waiters.*`, `webhook.processing.*`, `webhook.ingress.*`) for charting and alerting on queue capacity and webhook rate limits.

— Names metric fields and alert logic, no exact query sample v4.8.2

07 **Auto-completing run subscriptions** IMPROVED

69

By default the `for await` loop on run subscriptions now exits automatically on the final status update; pass `stopOnCompletion: false` to `runs.subscribeToRun` to keep the subscription open after a run finishes.

– Named API and parameter but no code sample v4.8.2

THINNER COVERAGE BELOW

08 **Dashboard query API requires token delegation** BREAKING 58

Query dashboard APIs now reject run tokens and public access tokens without user delegation; existing API key access, including restricted keys with the required query scopes, and MCP dashboard access via delegated tokens for current organization members remain supported.

– Explains access change but no migration steps shown v4.8.2

09 **Trace navigation and progressive loading** IMPROVED 52

Selecting a span that started a session now shows a link to that session in the span panel so you can jump from an orchestrator's run to its sub-agents; the run trace API now supports paging large traces, and the run page loads large traces progressively while updating live runs incrementally.

– Describes UI/API behaviour without naming the endpoint v4.8.2

10 **Ignored paths for Git deployments** NEW 50

A new Ignored paths setting in Git settings stops pushes and pull-request updates that only touch matching file patterns from creating a deployment; this does not apply when the Vercel integration is connected.

– Clear UI location but no example pattern given v4.8.2

11 **Experimental Convex integration** NEW 47

An experimental Convex component triggers tasks transactionally and reads their status and results through reactive queries, without configuring webhooks.

– Describes capability but no API names or setup steps v4.8.2

└──▷ **BREAKING ON UPGRADE**

! Query dashboard APIs now reject run tokens and public access tokens without user delegation; existing API key access (including restricted keys with the required query scopes) and MCP dashboard access via delegated tokens for current organization members remain supported.

! The legacy `clickhouse` event store is retired: task events now go to `clickhouse_v2`. Self-hosted deployments using `EVENT_REPOSITORY_DEFAULT_STORE=clickhouse` must switch the value to `clickhouse_v2` and copy older events per the self-hosting guide, or those events will not appear.

WAS THIS USEFUL? 👍 👎

thclaws

Feedback ⓘ v0.148.0 NOTES ↗ RANK

Open-source AI agent harness in native Rust — GUI, CLI, headless, and webapp from one binary. Multi-provider, MCP, skills, plugins, agent teams.

> thClaws adds Requesty as a new OpenAI-compatible provider option.

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

01 **Requesty provider support** NEW 35

thClaws now supports Requesty as an OpenAI-compatible provider alongside existing providers.

— Names provider but no setup steps or config details v0.148.0

WAS THIS USEFUL? 👍 👎

octo-cli

Feedback ⓘ v0.19.5 NOTES ↗ RANK

Metadata-driven CLI for AI Agent Bots — 48 operations across 7 domains, structured JSON envelope I/O, zero interactive prompts.

> octo-cli added project restore commands and expanded its marketplace domain with version and dependency commands, including the ability to pin relations to target versions.

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

01 **Marketplace version and dependency commands** NEW 55

The `marketplace` domain now adds version and dependency commands, and supports pinning relations to target versions.

— Names domain and commands but no exact subcommand syntax given. v0.19.5

02 **Project restore commands** NEW 15

octo-cli adds project restore commands.

— Mentioned only in summary with no detail on usage or mechanism. v0.19.5

WAS THIS USEFUL? 👍 👎

openexecutive

Feedback ⓘ v0.6.7 NOTES ↗ RANK

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

> openexecutive v0.6.7 brings live inbox monitoring and email reply into the Home Right now view, adds a pushback-response model to the orchestrator, and introduces host/user sign-in mismatch warnings.

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

01 **Delegation reply cards as conversation threads** IMPROVED 55

openexecutive now opens delegation reply cards as a conversation thread and automatically redrafts them when newer mail arrives.

— Explains redraft trigger but no config surface named v0.6.7

02 **Inbox monitoring with live alerts in 'Right now' view** NEW 52

openexecutive now shows the user's own inbox in the Right now view and sends a message/alert when it changes.

— Names the view and trigger but no further mechanism v0.6.7

03 **Host/user sign-in mismatch warning** NEW 45

openexecutive now warns when the host's user does not match the signed-in user and lets hosts see who is currently signed in.

— Clear behaviour described, no config or flag named v0.6.7

04 **Email reply from Home for 'Right now' points** NEW 40

openexecutive now lets users open and reply to Right now email points directly from the Home screen.

— UI path given, little mechanism detail v0.6.7

05 **Pushback-response model in orchestrator** NEW 40

openexecutive's orchestrator now answers using a pushback-response model after the recipient pushes back on a message.

— Names the model but lacks mechanism or scope detail v0.6.7

WAS THIS USEFUL? 👍 👎

atomic-agent

Feedback ⓘ v0.6.13 NOTES ↗ RANK

Atomic Agent is a local-first AI agent. Runs open-weight models on your own machine via llama.cpp.

> Atomic Agent gained opt-in control over the user's own Chrome profile and macOS desktop apps, folder-level trust gating for project skill files, and a new desktop file side panel with rich reply formatting and inline screenshots.

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

01 **Live agent control of Chrome and macOS apps** NEW 91

Setting `browser.connect` to `user-chrome` lets the agent control your already signed-in Chrome — working in its own tabs, asking once per site before opening or reading it, re-attaching automatically after Chrome restarts, and masking password, code and card fields from the model — while setting `tools.screen.enabled` lets the agent take screenshots, click and type in macOS apps beyond the browser, with browser windows remaining look-only and terminal windows accepting clicks but not typed input.

Use the agent inside your already-signed-in Chrome so it can act on sites where you are already authenticated.

yaml

```yaml
browser.connect: user-chrome
```

Let the agent click and type in macOS apps (not just the browser) to automate GUI workflows.

yaml

```yaml
tools.screen.enabled: true
```

— Names the config key and mechanism in full, no limits given Config key and exact scope limits named

v0.6.13

02 **File side panel in desktop app** NEW 83

Open a side panel with ⌥⌘F to browse every file associated with the current chat — files the agent created or mentioned, attachments, images, and screenshots — with preview support for text, code, Markdown, images, PDF, and CSV.

— Exact shortcut, scope and preview formats named desktop-v0.0.7

03 **Per-folder trust for project skills** NEW 75

Project files `AGENTS.md` , `CLAUDE.md` and installed project skills now load only after you explicitly trust the containing folder, once per folder; every file of every installed skill is scanned, and trust can be reviewed or changed later in desktop Settings › Skills.

— Mechanism and UI path given, no size limits `desktop-v0.0.7` `v0.6.13`

04 **Agent tool handling improvements** IMPROVED 60

The `os.shell.run` tool now accepts commands in the shapes models actually send rather than requiring strict formatting, and explains when a background job has ended instead of continuing to poll it, while the `os.fs.edit` tool now tolerates whitespace, indentation and quote drift when matching edits and suggests where a missing path actually is when the given path doesn't exist.

— Named tool and behaviour change, no runnable example Named tool and behaviour change, no example given

`v0.6.13`

THINNER COVERAGE BELOW

05 **Secrets masking and confirmation prompts** NEW 58

Secrets are now masked in tool output before the model reads them, and the agent asks for confirmation before reading its own secret files or making HTTP requests that send data.

— Behaviour described, no config surface named `v0.6.13`

06 **Rich reply formatting blocks** IMPROVED 57

Agent replies now render tables as scrollable, copyable tables, and give steps, key-value pairs, comparisons, callouts and file cards each their own distinct block.

— Names each block type, no interaction detail `desktop-v0.0.7`

07 **Page screenshots and compact view for small models** NEW 45

Adds screenshots of a page and a compact page view limited to interactive parts and changes, intended to help small local models work with browser pages.

— Described in prose, no config key or limits `v0.6.13`

08 **verify.syntax skips Windows stubs** IMPROVED 43

The `verify.syntax` check now skips the Windows `python3` and `bash` stubs, avoiding false failures caused by those stub binaries.

— Named check and stubs, but thin on mechanism `v0.6.13`

09 **Inline screenshots in chat** NEW 37

Screenshots the agent takes now appear inline as images in the chat, which can be opened or saved.

— Brief description, no further mechanism given `desktop-v0.0.7`

10 **DRY anti-repeat sampling for llama.cpp** NEW 30

Adds optional DRY anti-repeat sampling for llama.cpp engines.

— One-line mention with no config or mechanism detail `v0.6.13`

11 **Import Claude Code chats** NEW 28

Import Claude Code chats into one folder per project.

— Single-line description with no mechanism `desktop-v0.0.7`

WAS THIS USEFUL? 👍 👎

langchain4j

Feedback ⓘ 1.23.0 NOTES ↗ RANK

LangChain4j is an idiomatic, open-source Java library for building LLM-powered applications on the JVM. It offers a unified API over popular LLM providers and vector stores, and makes implementing tool calling (including MCP support), agents and RAG easy.

> langchain4j 1.23.0 expands its decision-agent framework with new model classes and prompt caching, adds a Google GenAI text-to-speech model and further Google GenAI integration capabilities, and introduces an Azure DocumentDB embedding store alongside new observability and LLM-judge testing tools.

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

01 **Decision agents and decision model builders** NEW 82

Adds an `OllamaDecisionModel` class with image input support, and adds decision agents with typed `DecisionInput`, conversation input support, and from(...) factory methods on all decision model builders. Also adds `promptCacheKey` to OpenAI decision models to enable prompt caching, and adds custom body parameters to all decision model builders.

— Names classes, fields and factory methods but no runnable example 1.23.0

THINNER COVERAGE BELOW

02 **Google GenAI integration capabilities** IMPROVED 52

The Google GenAI integration gains code execution, raw/recursive JSON schemas, multimodal tool results, async chat, and streaming cancellation support.

— Lists several additions without specific class or API names 1.23.0

03 **Azure DocumentDB embedding store** NEW 50

Adds an Azure DocumentDB embedding store as an alternative to the now-deprecated Azure Cosmos DB Mongo vCore store.

— Names the store and its deprecated predecessor, no config detail 1.23.0

04 **AiServiceInvocationSummaryEvent for observability** NEW 50

Adds an `AiServiceInvocationSummaryEvent` that aggregates all observability events from a single AI Service invocation into one object.

— Named class but no usage path shown 1.23.0

05 **LLM judge assertions in langchain4j-test** NEW 48

Adds LLM judges and assertions on chat requests in the `langchain4j-test` module for evaluating model outputs in tests.

— Names module but no concrete API or assertion syntax `1.23.0`

---

06 **GoogleGenAiTextToSpeechModel for text-to-speech** NEW

45

Adds a `GoogleGenAiTextToSpeechModel` class to convert text to speech using Gemini models.

— Named class and provider but no config or usage example `1.23.0`

---

07 **Agentic pattern activation conditions and streaming summarization**

IMPROVED

35

Allows any agent or agentic pattern to declare an activation condition, and allows agentic context summarization to use a `StreamingChatModel` in addition to a blocking model.

— Describes capability generically without naming the activation mechanism `1.23.0`

---

08 **GuardContent placement in Bedrock integration** NEW

33

Adds `guardContent` placement support to the Bedrock integration.

— Single named field with no further mechanism given `1.23.0`

WAS THIS USEFUL? 👍 👎

claw-orchestrator

Feedback ⓘ v7.16.0 NOTES ↗ RANK

Run Claude Code, Codex, Antigravity, Grok Build and OpenCode behind one runtime, and a run only counts as done when checks it didn't write pass. Cross-engine fan-out, councils, session handoff, spend caps, MCP server, ACP agent.

> claw-orchestrator v7.16.0 adds Claude Code as a selectable engine for goal nodes and introduces per-agent permission and sandbox modes for fanout agents.

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

01 Claude Code engine for goal nodes NEW

88

Goals can now run against Claude Code instead of Codex by passing `--engine claude` to `clawo goal`, or by setting `engine: 'claude'` on a `goal` node or on a `workflow_start`'s `goal` template; Codex remains the default engine when unset.

Run a goal against Claude Code in a one-off invocation to use Claude Code's `/goal` instead of a Codex thread.

```
$ clawo goal --engine claude 'Refactor the auth module to use JWT'
```

Set Claude Code as the engine for a goal node in a workflow definition so every run of that node uses Claude Code's `/goal`.

```yaml
goal:
  engine: 'claude'
  template: 'Audit all API endpoints for missing auth checks'
```

— Names exact flag, config key and default behavior with runnable examples v7.16.0

THINNER COVERAGE BELOW

02 Permission and sandbox modes for fanout agents NEW

40

`fanout_start` agents now declare `permissionMode` and `sandboxMode` per agent.

— Names the fields but gives no values, mechanism or example v7.16.0

WAS THIS USEFUL?  

truefoundry TrueForge

Feedback ⓘ

@truefoundry/trueforge-ui@0.181.0 NOTES ↗ RANK

The open-source agent harness - the runtime layer that turns an LLM into a working agent.

> TrueForge's 0.181.0 release wave adds new runtime observability (context-compaction lifecycle events, expanded Prometheus metrics), new API fields for models, schedules and sessions, a Keenable web-search MCP preset, and tightens security with SSRF redirect blocking and admin-only sandbox environment creation, alongside a round of session, schedule and agent-config UI improvements.

└──▷ **WHAT SHIPPED · 17 FEATURES** most completely described first ? what's the number?

01 **API response field additions for models, schedules and sessions**

NEW 90

`GET /api/v1/models` now returns a `warning` field on each model when reasoning effort or context length is not configured; schedule run responses now include a `session_id` field (backfilled for existing runs, `null` when a run has no session); and session responses now include a `last_turn_details` field (`{ id, state: { status } }`, null before the first turn, kept in sync as the last turn changes status, with existing sessions backfilled).

Check whether a model has misconfigured reasoning effort or context length before wiring it into an agent.

```
$ curl -s 'https://<your-trueforge-host>/api/v1/models' \
  -H 'Authorization: Bearer <token>' | jq '.[] | select(.warning
!= null) | {id, warning}'
```

Check the status of the most recent turn on a session without fetching the full turn list.

```
$ GET /sessions/{session_id}

Response includes:
{
  "last_turn_details": {
    "id": "<turn_id>",
    "state": { "status": "completed" }
  }
}
```

— Named endpoint, exact field shapes, and runnable curl/GET examples.

`@truefoundry/trueforge@0.181.0` `@truefoundry/trueforge-core@0.181.0`

---

02 **Context compaction lifecycle events** NEW 70

Emit `thread.context_compaction_started` and `thread.context_compaction_finished` events around context compaction, carrying `trigger`, `usage`, and `duration_ms` so integrators can measure compaction cost and timing.

Subscribe to context compaction lifecycle events in your runtime integration to measure compaction cost and duration.

> 📍 Listen for `thread.context_compaction_started` and `thread.context_compaction_finished` events on the thread; each `finished` event includes `trigger`, `usage`, and `duration_ms`.

— Named events and fields with a runnable listener example.

`@truefoundry/trueforge-ui@0.181.0`

---

03 **Per-call LLM headers and metadata** NEW 65

Support per-call LLM `headers` and `metadata` on TrueFoundry-mode model invokes, with metadata folded into the `x-tfy-metadata` header and compaction calls stamped with `tfg.compaction: true` for trace visibility.

— Named header and metadata key, no direct usage example. `@truefoundry/trueforge@0.181.0`

---

04 **Model capability defaults for reasoning effort and web search** IMPROVED 65

Reasoning effort now defaults to `medium` in the New Chat composer and Agent Config when the selected model offers it, falling back `medium` → `high` → `low` when it does not; Agent Config always shows max tokens and reasoning effort with playground fallbacks when the model doesn't advertise them; and web search stays enabled by default in New Chat when the capability is available.

— Explains fallback logic and scope across composer and Agent Config.

`@truefoundry/trueforge@0.181.0` `@truefoundry/trueforge-ui@0.181.0`

05 **Session view timeline and metrics enhancements** IMPROVED

60

The session view now shows a token breakdown donut next to the session Tokens metric, the model name in each turn header, each turn's created and completed time in the timeline hover, sub-agent model time between tool-only replies on the timeline, and lets the timeline chart scroll horizontally when a session has many turns.

— Several concrete UI changes but navigation-only actionability.

`@truefoundry/trueforge-ui@0.181.0`

06 **Admin-only sandbox environment creation** BREAKING

60

Sandbox environment creation is now admin-only, and list-permissions exposes `MANAGE` and `DELETE` (plus tenant `sandbox_environment` `CREATE`) so the UI matches the API.

— Named permission values but no migration steps given.

`@truefoundry/trueforge-assistant-ui-run…`

THINNER COVERAGE BELOW

07 **Denser Agents library rows with metrics** IMPROVED

55

Agents library rows now show creator nested under the name, a two-line configuration, and per-page sessions, spend, 14-day sparkline, and last-run when metrics are available.

— Names specific metrics shown, UI path only. `@truefoundry/trueforge-ui@0.181.0`

08 **Prometheus metrics expansion** IMPROVED

50

Prometheus scrape now records outbound HTTP, model calls, MCP, harness limits, tool-call duration, active threads, turn lifetime, parallel web search, ServiceFoundry requests, Postgres query duration, and Daytona and Redis SDK meters.

— Lists meters covered but no scrape endpoint or setup detail.

`@truefoundry/trueforge@0.181.0`

09 **Monaco markdown editor for agent instructions** NEW

50

Edit agent instructions in a Monaco markdown editor with line numbers, folding, and a full-screen mode.

— Names the editor and features, clear UI entry point. `@truefoundry/trueforge-ui@0.181.0`

10 **Keenable web search MCP preset** NEW

45

Add a keyless Keenable remote MCP server preset to the shipped MCP catalog, giving users an opt-in web search option alongside existing presets.

— Names the preset and catalog but no config detail. `@truefoundry/trueforge@0.181.0`

11 **Blocked large tool response messages show size and limit**

IMPROVED 45

Blocked large tool response messages now state the exact content size and which limit blocked it, in characters, and the file-dump message prompts the model to plan reads against the total.

— Describes message content but not the limit values themselves.

`@truefoundry/trueforge-core@0.181.0`

12 **Custom cron scheduling** NEW 40

Allow entering a custom cron expression when creating or editing a schedule.

— Clear capability but no detail on cron syntax or limits.

`@truefoundry/trueforge-ui@0.181.0`

13 **Schedule run opens linked session** NEW 40

Clicking a schedule run on the Schedules page now opens the linked session directly in the Sessions page.

Navigate to the linked session directly from a schedule run to investigate what happened during a specific scheduled execution.

📍 In TrueForge, go to Schedules, find the schedule run you want to inspect, and click the run entry — TrueForge opens the linked session in the Sessions page.

— Clear navigation path with a step-by-step example. `@truefoundry/trueforge-ui@0.181.0`

14 **SSRF-guarded fetch blocks HTTPS-to-HTTP redirects** IMPROVED 40

Block HTTPS-to-HTTP redirects in the outbound SSRF-guarded fetch so cleartext downgrades cannot follow a secure hop.

— Clear security behavior but no reproduction or config detail.

`@truefoundry/trueforge-core@0.181.0`

15 **Deferred tool instructions show descriptions for lazy-loaded MCP servers**

IMPROVED 35

MCP servers configured with `preload: false` now show their saved description before tools are listed in deferred tool instructions.

— Names the config key but limited mechanism detail. `@truefoundry/trueforge@0.181.0`

16 **Read-only shared session view** NEW 35

Route shared-session viewers to a read-only session view and hide in-flight turns.

— Behavior described but no scope or edge cases. `@truefoundry/trueforge-ui@0.181.0`

17 **Sandbox environment status badge icons** IMPROVED 25

Show check and triangle-alert icons on sandbox environment Ready and Failed status badges.

— Minor UI cosmetic change with little mechanism. `@truefoundry/trueforge-ui@0.181.0`

WAS THIS USEFUL? 👍 👎

platypus

Feedback ⓘ v3.29.0 NOTES ↗ RANK

Self-hosted AI Agents for your whole team — on your infrastructure, your models, around the clock.

> platypus added a Daytona Sandbox backend plugin and a new Kanban Boards Tool set for managing boards and columns, with incremental metadata improvements to board state retrieval.

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

01 **Kanban Boards Tool set for board and column management** NEW 55

platypus introduces a Kanban Boards Tool set for managing boards and columns. The `getBoardState` card summaries returned by this tool set now include an `updatedAt` field.

— Names tool and a specific field, but no further detail `v3.29.0`

02 **Daytona Sandbox backend plugin** NEW 25

platypus adds a Daytona Sandbox backend plugin, expanding the set of sandbox backends available to agents.

— Only a bare name, no mechanism or config given `v3.29.0`

WAS THIS USEFUL? 👍 👎

agent-app-orchestrator

Feedback ⓘ v5.3.0 NOTES ↗

RANK

Agent orchestration service for the Agent Landing Zone, built on Microsoft Foundry Agent Service and Microsoft Agent Framework. Coordinates specialized agents for conversational experiences with grounded knowledge, NL2SQL and MCP tools.

> agent-app-orchestrator ships native per-user Search permissions with forwarded credential support across its core endpoints, alongside tightened handling of downstream authentication claims.

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

01 **Forwarded Search credentials and native per-user permissions**

NEW 83

Both `/responses` and `/invocations` now accept an explicit forwarded Search credential for official hosted forwarded credential flows; Work IQ uses its own separately registered application assertion, and platform partition ids, request-body identity, and Search tokens do not substitute for that assertion. Native per-user Search permissions forward the delegated Search credential to native permission enforcement for indexed and remote knowledge sources, preserving conversation-upload scope without requiring manual group authorization or blob download access.

– Names exact endpoints and credential mechanism, no code sample v5.3.0

THINNER COVERAGE BELOW

02 **Bounded downstream claims recovery for authentication failures**

IMPROVED 56

Claims recovery now passes only canonical, validated claims through the existing failed-turn contract in both hosted adapters; ambiguous or malformed challenges no longer initiate claims recovery, and authentication failures are treated as non-retryable.

– Describes mechanism and scope but no concrete API or config surface v5.3.0

WAS THIS USEFUL? 👍 👎

persona

Feedback ⓘ @runtypelabs/persona@4.33.0 NOTES ↗ RANK

Create agentic front-end experiences for the web in VanillaJS. Lightweight, extensible, and WebMCP-native.

> Persona shipped a major release adding opt-in dictation surfaces, durable background conversations, frameless inline embeds, and live widget relocation via moveTo(), alongside deep tool-call and reasoning display customization, a new operator role for human replies with a modelMessageTransform hook, programmatic tool-call reveal via controller.revealToolCall(), and an exported preprocessStreamingMarkdown helper for custom renderers.

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

01 Programmatic tool-call reveal and highlight NEW 98

controller.revealToolCall(toolCallId, { expand?, highlightMs? }) programmatically opens, centers, and highlights any tool call in the transcript; tool bubbles now carry a `data-tool-call-id` attribute so hosts can map clicks back to their own records. Reveal highlights survive streaming updates for their configured duration, repeated reveals restart the highlight timer, revealing a tool call pauses automatic scroll-following even when centering reaches the bottom, and a later reveal cancels pending centering from the previous call.

Jump to and highlight a specific tool call in the transcript when a user clicks a related item in your own UI — mapping your record's ID to the widget's transcript entry.

```javascript
controller.revealToolCall(toolCallId, { expand: true,
highlightMs: 2000 });
```

— Runnable API with example and documented edge-case behavior.

@runtypelabs/persona@4.32.0

02 Tool call and reasoning display customization NEW 90

`features.toolCallDisplay.groupSummaryMaxItems` (default 3, set to zero for a count-only aggregate) caps distinct summary categories in homogeneous tool-group headers; `features.toolCallDisplay.showToolName` opts in to showing raw tool names above tool details (hidden by default in V5); `toolCall.iconName` lets per-tool registered icons override status and visibility controls; and grouped tool children gain an inset line independent of stack or collapsible layout (V5 groups use inset by default, with plain indentation available as an override).
`features.reasoningDisplay.maxHeight` sets a custom or unlimited height cap on

expanded reasoning with keyboard scrolling and live resets; shared `activity.panel` background, border color/width, radius, and padding tokens apply per-property tool and reasoning panel overrides; opt-in shared height transitions for tool, tool-group, and reasoning disclosures default to 160ms with streaming-safe rendering and reduced-motion support; and disclosure chevrons are configurable, defaulting to right/down in V5 while retaining V4 directions. Canonical `layout` controls for tool and reasoning displays preserve existing `variant` aliases, and the disclosure body gains plain, inset, and panel reasoning detail options.

Cap reasoning disclosure height and enable shared activity overflow fades so expanded tool and reasoning panels scroll rather than grow unbounded.

javascript

```javascript
persona.configure({
  features: {
    reasoningDisplay: {
      maxHeight: 320
    },
    toolCallDisplay: {
      showToolName: false,
      groupSummaryMaxItems: 3
    }
  },
  activity: {
    panel: {
      background: '#1a1a2e',
      borderColor: '#3a3a5c',
      borderWidth: '1px',
      radius: '8px',
      padding: '10px 12px'
    }
  }
});
```

— Many config keys named, with a runnable configure() example.

`@runtypelabs/persona@4.30.0`

---

03 **Streaming Markdown repair export** NEW 80

The `preprocessStreamingMarkdown` function is now exported so custom message renderers can apply the same incomplete-Markdown display repairs as the built-in renderer, stabilising incomplete emphasis, inline code, links, and images while a streaming response is still arriving.

Use `preprocessStreamingMarkdown` in a custom renderer to stabilise incomplete emphasis, inline code, links, and images while a streaming response is still arriving.

```javascript
import { preprocessStreamingMarkdown } from
'@runtypelabs/persona';

function myMessageRenderer(partialText) {
  const safeText = preprocessStreamingMarkdown(partialText);
  // render safeText to the DOM
}
```

— Exported function with runnable example, otherwise thin. `@runtypelabs/persona@4.31.0`

04 **moveTo() for relocating live widgets** NEW 80

widget.moveTo() relocates a live direct-layout widget between connected containers in the same document while preserving its conversation, composer, renderers, and streaming session.

Relocate a live embedded widget to a different DOM container (for example when a layout panel is dragged or docked) without losing the ongoing conversation or session.

```javascript
const widget = persona.install({ /* existing config */ });
// Later, when the host container changes:
widget.moveTo(document.getElementById('new-panel-container'));
```

— Runnable method with example showing container relocation. `@runtypelabs/persona@4.30.0`

05 **Operator role for human replies in transcript** NEW 75

A new `operator` UI role lets human replies appear on the assistant side, with its own theme tokens, accessible author label, optional structured author provenance, custom message actions, and attachment-only reply support. A synchronous `modelMessageTransform` hook lets stateless adapters convert these human operator contributions into attributed model context in chronological order, and operator messages are kept out of model requests by default for backends that own their own conversation history, with provenance preserved through injection, custom history, and persistence.

— Role and hook named but no usage example given. `@runtypelabs/persona@4.33.0`

06 **Composer overflow menus, attachment previews, and tooltips** NEW 70

Space-aware action overflow adds themeable menus with configurable width, placement, trigger icons, descriptions, group dividers, and menu-only labels. Image and

document attachment previews gain filename chips, file icon overrides, row layouts, per-kind remove styling, and a managed preview hook. Accessible tooltips for icon-only composer actions support custom text and opt-out, with a 400ms hover delay.

— Several composer surfaces named but no config keys or example.

`@runtypelabs/persona@4.30.0`

---

07 **Dictation transcript and waveform surfaces** NEW 65

Opt-in dictation transcript and waveform surfaces add cancel, finish, send, and review controls, lifecycle events, and a managed `renderDictation` hook; optional voice bars respond to live input levels and reduced-motion preferences.

— Named hook but no usage example provided. `@runtypelabs/persona@4.30.0`

---

08 **Durable background conversations** NEW 62

Opt-in durable background conversations run through an explicit backend adapter with scoped execution status and read-catch-up from matching checkpoints, plus per-conversation drafts, attachments, and reading-position preservation, with a stop policy for switching conversations during a run.

— Mechanism described but no adapter API or example shown. `@runtypelabs/persona@4.30.0`

---

09 **Whole-turn activity disclosure** NEW 60

An opt-in whole-turn activity disclosure classifies progress and final messages explicitly, exposes a synchronous host adapter and a custom header hook, and preserves per-tool descriptive action titles through a synchronous host title resolver independent of turn classification, exposed to card summary renderers.

— Mechanism described but no config key or example given. `@runtypelabs/persona@4.30.0`

THINNER COVERAGE BELOW

10 **Configurable conversation-open loading skeleton** NEW 50

`loadingIndicator.conversationOpenVariant` chooses skeleton placeholders when opening conversations, with transcript fade-in and a delayed centered spinner.

— Named config key, no default or example shown. `@runtypelabs/persona@4.30.0`

---

11 **Frameless inline embeds** NEW 30

Frameless inline embeds support shrinkable full-height layouts and scoped transcript, fade, and composer-footer surfaces.

— Single sentence, no config keys or example. `@runtypelabs/persona@4.30.0`

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Providers & GPU Clouds

Nebius AI Cloud

Feedback ⓘ changelog · Oct 30, 2026

CHANGELOG ↗ RANK

Nebius AI Cloud provides GPU compute, managed Kubernetes, distributed training infrastructure, and serverless AI workloads.

> Nebius AI Cloud added public spot GPU pricing transparency and announced several breaking policy changes affecting standalone VM products, Kubernetes SLA terms, and support ticket priorities.

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

01 **Public spot GPU pricing and 30-day history** NEW 68

Current spot prices and 30-day price history for preemptible GPU platforms are now viewable publicly on the Nebius website or via the API without an account, and in the web console or CLI with an account, to help choose a region or set a max price for a pricing policy.

– Explains access channels and use case but no API endpoint named

changelog-20261030-0494998a

02 **Standalone applications and Container VMs discontinued** 62

BREAKING

Standalone applications and Container VMs are no longer available; third-party software such as JupyterLab, ComfyUI, and Qdrant must now be deployed on Managed Service for Kubernetes clusters instead.

– Names affected products and replacement path but no migration steps

changelog-20261030-0494998a

THINNER COVERAGE BELOW

03 **Kubernetes SLA excludes EOL versions and modified add-ons** 54

BREAKING

Starting November 2, 2026, the Managed Service for Kubernetes SLA does not apply to clusters running an end-of-life Kubernetes version or to unavailability caused by modifying preinstalled add-ons outside the public API.

– Clear policy change with date but no remediation guidance changelog-20261030-0494998a

04 **Support ticket priority levels reduced** BREAKING 52

Starting November 2, 2026, support ticket priorities change to Medium (4-hour first response) and Low (1 business day); customers with a committed volume contract signed by November 1, 2026 keep current priorities for the contract duration.

— Specific response times and grandfather clause given, no further action

changelog-20261030-0494998a

05 **Restart support for Serverless AI endpoints** NEW 18

Nebius AI Cloud added restart support for Serverless AI endpoints, per the changelog summary.

— Only named in summary, no mechanism or detail given changelog-20261030-0494998a

└──▷ **BREAKING ON UPGRADE**

! Standalone applications and Container VMs are no longer available; third-party software such as JupyterLab, ComfyUI, and Qdrant must now be deployed on Managed Service for Kubernetes clusters.

! Starting November 2, 2026, the Managed Service for Kubernetes SLA does not apply to clusters running an end-of-life Kubernetes version or to unavailability caused by modifying preinstalled add-ons outside the public API.

! Starting November 2, 2026, support ticket priorities change to Medium (4-hour first response) and Low (1 business day); customers with a committed volume contract signed by November 1, 2026 keep current priorities for the contract duration.

WAS THIS USEFUL? 👍 👎

Crusoe Cloud

Feedback ⓘ updated · Oct 30, 2026 NOTES ↗ RANK

Crusoe Cloud provides GPU infrastructure and managed inference for AI training and deployment.

> Crusoe Cloud added VM disk I/O and NVMe drive health metrics accessible via multiple surfaces, while discontinuing A100 availability in one region.

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

01 **VM disk I/O and NVMe health metrics** NEW 62

VM Telemetry now exposes disk I/O and NVMe drive health metrics, accessible via the `API`, Telemetry Conduit, and Crusoe MCP.

– Names the three access surfaces but no endpoint details. snapshot-20261030

THINNER COVERAGE BELOW

02 **A100 instances removed from us-west1a** BREAKING 35

A100 instances are no longer available in `us-west1a`.

– States the change but no migration guidance. snapshot-20261030

└──▷ **BREAKING ON UPGRADE**

! A100 instances are no longer available in `us-west1a`.

WAS THIS USEFUL? 👍 👎

CoreWeave

Feedback ⓘ updated · Oct 30, 2026 NOTES ↗ RANK

CoreWeave is an AI cloud platform offering GPU compute, managed Kubernetes and storage for training and inference.

> CoreWeave introduced Kata Containers runtime selection on CoreWeave Kubernetes Service (CKS) to isolate GPU workloads in a virtual machine, now available in private preview.

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

01 **Kata Containers runtime isolation on CKS** NEW 55

Workloads on CoreWeave Kubernetes Service (CKS) can now select the Kata Containers runtime to isolate GPU workloads inside a virtual machine. The feature is in private preview; customers must contact Support to request access.

— Names mechanism and access path but lacks config/flag details. snapshot-20261030

WAS THIS USEFUL? 👍 👎

RunPod

Feedback ⓘ New ReleaseTerraform provider 2.0.0 NOTES ↗

RANK

RunPod is an AI developer cloud platform for experimenting, training, fine-tuning, deploying, and scaling AI workloads across 31 global regions.

> RunPod's Terraform provider 2.0.0 is a major rework built on REST API v2, adding Instant Clusters, global volumes, CPU Pod support, and new v2 billing data sources, while deprecating several 1.x resources and fields that require a migration before upgrading.

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

01 Provider authentication and endpoint configuration

BREAKING

90

The 2.0.0 provider can authenticate via the `RUNPOD_API_KEY` environment variable or the `api_key` argument, target a non-default host with `base_url` or `RUNPOD_API_BASE_URL`, and point at the GraphQL shim layer with `graphql_url` or `RUNPOD_GRAPHQL_URL` for non-production environments. This is a breaking change for existing configs: `base_url` now points at the REST v2 host, so any custom 1.x value will misdirect requests, and the `rest_v1_url` argument used in 2.0.0 release candidates is deprecated and ignored.

Declare the 2.0.0 provider and authenticate via environment variable so the API key never appears in state or config files.

hcl

```hcl
terraform {
  required_version = ">= 1.6"
  required_providers {
    runpod = {
      source  = "runpod/runpod"
      version = "~> 2.0"
    }
  }
}

provider "runpod" {}
```

— Names every env var and argument with exact before/after behaviour

New ReleaseTerraform provider 2.0.0

02 v2 billing data sources replacing 1.x equivalents

BREAKING

85

New data sources `runpod_pod_billing`, `runpod_serverless_billing`, and `runpod_network_volume_billing` replace the deprecated 1.x equivalents `runpod_billing_pod`, `runpod_billing_network_volume`, and `runpod_billing_endpoint`; Terraform prints a warning on every plan that still touches one of the old data sources.

— Names old and new data sources with the deprecation warning behaviour

New ReleaseTerraform provider 2.0.0

---

03 **New Terraform resources for clusters, volumes, pods, SSH keys and registries**

NEW

80

Version 2.0.0 adds `runpod_cluster` for Instant Clusters, `runpod_global_volume` (with a matching data source) for global volumes, `runpod_pod_action` for one-off Pod actions such as a restart, `runpod_account_ssh_keys` for account SSH keys, and `runpod_registry_delegation` for delegating Amazon ECR access alongside the existing `runpod_registry` resource.

— Lists every new resource name but little mechanism per resource

New ReleaseTerraform provider 2.0.0

THINNER COVERAGE BELOW

04 **CPU Pod support in the runpod_pod resource** IMPROVED

45

The `runpod_pod` resource now covers CPU Pod types, so CPU Pods can be managed alongside GPU Pods through the same Terraform resource.

— Names the resource but gives no parameter or type values

New ReleaseTerraform provider 2.0.0

---

05 **Deprecated registries username field** DEPRECATED

40

The `registries[].username` field on `runpod_registries` is deprecated and now always returns null.

— States the deprecated field but no replacement or migration path

New ReleaseTerraform provider 2.0.0

└──▷ **BREAKING ON UPGRADE**

! Remove the `base_url` argument from your provider block if you set it: it now points at the REST v2 host and will misdirect any 1.x configuration that relied on a custom value.

! The `rest_v1_url` argument from the 2.0.0 release candidates is deprecated and ignored.

! The `registries[].username` field on `runpod_registries` is deprecated and always returns null.

! The 1.x billing data sources `runpod_billing_pod`, `runpod_billing_network_volume`, and `runpod_billing_endpoint` are deprecated; Terraform prints a warning on every plan that touches one.

! Upgrading from 1.x requires configuration and state changes — follow the migration guide before upgrading.

WAS THIS USEFUL? 👍 👎

◆ Agentic Infrastructure

Snowflake Cortex AI

Feedback ⓘ changelog · Oct 30, 2026

CHANGELOG ↗ RANK

Snowflake Cortex AI provides managed language-model functions, search, and agents that work with data governed in Snowflake.

> Snowflake Cortex AI's biggest addition is an AGENTIC mode for `AI_PARSE_DOCUMENT` that reasons across documents, alongside a new Coding Agent MCP tool, richer Cortex Agents configuration, code skills for interactive analytics, per-app usage tracking, and a regional restriction on fine-tuning.

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

01 **AGENTIC mode for AI_PARSE_DOCUMENT** NEW 80

`AI_PARSE_DOCUMENT` gains a new `AGENTIC` mode (set via `{'mode': 'AGENTIC'}`) that processes documents with intelligent, multi-step reasoning, now in Public Preview. It can reason across multiple pages of a document to extract and resolve fields such as party names, dates, and payment terms.

Extract structured data from a complex multi-page PDF using the new AGENTIC mode, which can reason across pages and resolve ambiguous fields.

```
$ SELECT AI_PARSE_DOCUMENT(
  TO_FILE('@my_stage/contract.pdf'),
  'Extract all party names, effective dates, and payment terms',
  {'mode': 'AGENTIC'}
) AS parsed;
```

— Named function and mode with a runnable example query. changelog-20261030-d7af6994

THINNER COVERAGE BELOW

02 **Richer Cortex Agents context and reasoning controls** IMPROVED 55

Cortex Agents now support a `thinking effort` parameter to control reasoning depth, plus `system instructions` and stage mounts to supply richer context to agents.

— Named parameters but no usage example or exact syntax. changelog-20261030-d7af6994

03 **Per-app Cortex AI usage tracking** NEW 45

Cortex AI consumption can now be tracked per Native App in the usage history view.

— Names the usage history view as a clear starting point. changelog-20261030-d7af6994

04 **Fine-tuning training restricted in AWS US East 1** BREAKING 45

Cortex Fine-tuning: new model training is no longer available in AWS US East 1.

— Names affected feature and region but no migration guidance.

changelog-20261030-d7af6994

05 **Coding Agent MCP tool** NEW 40

A new Coding Agent MCP tool lets Cortex's coding agent connect to external MCP-compatible clients and workflows.

— Names the tool but no setup steps or syntax given. changelog-20261030-d7af6994

06 **Cortex Code skills for interactive analytics** NEW 35

Snowflake Cortex AI adds Cortex Code skills for interactive analytics, enabling agents to generate and run code against live data.

— Described only in general terms, no mechanism or example. changelog-20261030-d7af6994

└──▷ **BREAKING ON UPGRADE**

! Cortex Fine-tuning: new model training is no longer available in AWS US East 1.

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore Feedback ⓘ

changelog · Oct 30, 2026 CHANGELOG ↗

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

AgentCore adds FUSE file system mounting, session context keys for scoped access, model fallback routing with circuit-breaker support, and a rebuilt CLI v1.0 with direct resource commands and a migration tool.

Mount file systems with FUSE inside agent containers on platform version V2 runtimes, letting agents treat mounted storage as local files without orchestrating downloads and uploads.

Scope access to a session or runtime using session context keys on platform version V2 agent runtimes, restricting sessions that share an execution role to only their own data with a single policy.

NEW INTEGRATION Run model fallback for inference routing in AgentCore Gateway by configuring a `matchModels` condition and an ordered fallback chain on the `routeToTarget` action, with per-destination first-byte timeouts and a service-owned circuit breaker toggled on by default.

Use AgentCore CLI v1.0 (`@aws/agentcore` 1.0.0) command groups `harness` , `runtime` , `memory` , `identity` , `gateway` , `payment` to work with AgentCore resources directly by ID or ARN, without a project.

Migrate existing v0 CLI projects to v1 with `agentcore migrate` , which shows a read-only migration plan, refactors the CloudFormation stack, keeps v0 physical resource names, and can resume an interrupted migration.

+1 more

NEW SUPPORTED PLATFORM AWS Agent Registry regional expansion

└──▷ BREAKING ON UPGRADE

! AgentCore CLI v1.0 (`@aws/agentcore` 1.0.0) has breaking changes relative to v0; commands that change a project, such as `add` and `remove` , require the v1 project format. Run `agentcore migrate` to upgrade.

WAS THIS USEFUL? 👍 👎

Perplexity API

Feedback ⓘ updated · Oct 30, 2026 NOTES ↗

RANK

Perplexity API provides language models with real-time web search and retrieval for building applications that research and answer questions.

> Perplexity's Agent API gains a new `ultra-preview` effort level for broad research, a step-budget ceiling raised to 500 with model and token updates across `low` / `high` / `xhigh` / `fast` tiers, and a breaking change rerouting the `wide-research` alias to `xhigh` while removing the undocumented `ultra` effort name.

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

01 Effort level configuration updates: step caps, tokens, and model swaps

IMPROVED 85

The `max_steps` ceiling is raised from 100 to 500 across all effort levels. `low` effort (`low@2026-10-24.0`) max output tokens rise from 32,768 to 128,000. `high` effort (`high@2026-10-28.0`) switches from `openai/gpt-6-sol` to `openai/gpt-6.1-terra` , raising max steps from 15 to 40 with a per-run cap of 40 tool calls. `xhigh` effort (`xhigh@2026-10-29.0`) switches to `anthropic/claude-opus-5-5` with `openai/gpt-6.1-sol` as fallback, raising max steps from 100 to 300 and starting context compaction at 150,000 tokens. `fast` effort (`fast@2026-10-16.0`) gains a fallback to `openai/gpt-5.6-luna` when `openai/gpt-6-luna` is unavailable. Frozen configurations for any effort level will no longer match these updated public API fields unless callers copy the new values manually.

— Every model, version tag and limit named but no single command `snapshot-20261030`

02 New `ultra-preview` effort level for Agent API NEW

75

Callers can set `effort="ultra-preview"` in Agent API calls to run long-running, broad research and data-gathering tasks that optimise for coverage over speed. This replaces the undocumented `ultra` effort name, which is removed and now returns an error if used.

Run a broad competitive-intelligence sweep where coverage matters more than latency, using the new `ultra-preview` effort level.

```
$ curl -s https://api.perplexity.ai/agent/run \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{
    "effort": "ultra-preview",
    "max_steps": 200,
    "messages": [{"role": "user", "content": "Survey the top 20
open-source SIEM projects released in 2025-2026 and summarise
their detection coverage."}]
  }'
```

— Named flag with a runnable example, but no numeric limits given `snapshot-20261030`

THINNER COVERAGE BELOW

03 **`wide-research` alias rerouted to `xhigh`, dropping system prompt**

45

BREAKING

The `wide-research` alias now selects the `xhigh` effort level instead of its previous target, so callers relying on prior `wide-research` behaviour should verify the new routing. Separately, `xhigh` no longer sends a system prompt in its frozen configuration, so callers that depended on those instructions must supply their own.

— Names the alias and consequence but no migration command `snapshot-20261030`

04 **`fetch_url` extraction model updated** IMPROVED

35

`fetch_url` content extraction now uses `openai/gpt-6-luna` for `low`, `medium`, and `high` effort levels.

— Names the model and tool but no behavioural detail `snapshot-20261030`

05 **Step budget reminders during agent runs** IMPROVED

30

`low`, `medium`, `high`, and `xhigh` effort levels now remind the model of its remaining step budget during a run.

— States the change but no mechanism or effect detail `snapshot-20261030`

└──▷ BREAKING ON UPGRADE

! The undocumented `ultra` effort name is removed and now returns an error; callers must switch to `effort="ultra-preview"`.

! The `wide-research` alias now selects `xhigh` instead of its previous target; callers relying on `wide-research` behaviour should verify the new routing.

! `xhigh` no longer sends a system prompt in its frozen configuration; callers that depended on those instructions must supply their own.

! Frozen configurations for any effort level will no longer match the updated public API fields unless callers copy the new values manually.

WAS THIS USEFUL? 👍 👎

Tracer Cloud OpenSRE

Feedback ⓘ v0.1.2026.10.30 NOTES ↗

RANK

OpenSRE is an open-source toolkit for building AI SRE agents.

> OpenSRE's only update in this window lets users manage third-party provider model connections directly through the CLI login flow.

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

01 **CLI login for provider model connections** NEW 35

Third-party provider model connections can now be managed directly through the CLI login flow.

— Names CLI login flow but no exact command or flag given. v0.1.2026.10.30

WAS THIS USEFUL? 👍 👎

Hugging Face Transformers

Feedback ⓘ v5.22.0

NOTES ↗

Hugging Face Transformers is a model-definition framework for machine learning models in text, vision, audio, and multimodal tasks, for both inference and training.

Transformers v5.22.0 enables adapter training on FP8-quantized weights with chunked MoE backward cutting peak memory from ~28.9 GiB to ~2.8 GiB at 8k tokens.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v5.22.0
https://github.com/huggingface/transformers.git
# already have the repo? check out this version:
$ git checkout v5.22.0
```

Train LoRA and other PEFT adapters directly on top of FP8-quantized weights; `is_trainable` is now `True` for FP8 layers, and the MoE layer backward pass now dequantizes experts in chunks, cutting peak memory from ~28.9 GiB to ~2.8 GiB at 8k tokens.

NEW INTEGRATION Mistral Large 4 model support

NEW INTEGRATION Nemotron 3.5 Transcribe model support

└──▷ BREAKING ON UPGRADE

! Expert-parallel (EP) logic has been separated into its own file (#49679); any code importing EP symbols from their previous location will break.

WAS THIS USEFUL? 👍 👎

emisar

Feedback ⓘ v0.54.0 NOTES ↗ RANK

Emisar lets AI agents securely run infrastructure actions through MCP, with approval controls that protect production systems.

> Emisar's v0.54.0 release focuses on making MCP approval requests reliably reach users—including through subagents—alongside a reorganized console sidebar, a more usable policy editor, and a breaking BunnyCDN pack update.

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

01 MCP approval-link notifications reach users reliably

IMPROVED

65

When an MCP action or runbook waits for approval, server instructions and every approval result now tell the agent to notify the user immediately with the exact approval link, fire once per request, continue independent work, and never treat a chat reply as approval. An MCP agent running as a subagent now forwards approval notices to its parent agent so the notification reaches the user rather than stopping in the subagent's reply.

— Explains mechanism and subagent fix but no config surface named v0.54.0

02 BunnyCDN pack 0.1.11 rebuilds edge rules and redacts secrets

65

BREAKING

BunnyCDN pack 0.1.11 now rebuilds each Pull Zone edge rule from known fields and redacts header, cookie, query-string, and credential values; earlier versions of the pack are retired and no longer supported.

— Names version, mechanism and redacted fields but no migration steps v0.54.0

THINNER COVERAGE BELOW

03 Console sidebar reorganized into Connect, Operate, Control

IMPROVED

50

The console sidebar is now grouped into Connect, Operate, and Control sections: service accounts sit with runners and AI agents, runbooks with runs, and policy, packs, team, and billing together.

— Clear UI reorganization but no deeper mechanism v0.54.0

04 Policy editor usability improvements

IMPROVED

50

The policy editor now keeps targeted rulesets open, hides approval settings that cannot apply, searches ruleset targets in one dropdown, and starts each ruleset's preview beside its editor.

— Names four concrete UI behaviors but no config keys v0.54.0

└──▷ BREAKING ON UPGRADE

! BunnyCDN pack versions earlier than 0.1.11 are retired and no longer supported.

WAS THIS USEFUL? 👍 👎

Google Gen AI JavaScript SDK

Feedback ⓘ v2.32.0

NOTES ↗

The Google Gen AI JavaScript SDK provides TypeScript and JavaScript interfaces to Gemini APIs and live model sessions.

Google Gen AI JavaScript SDK adds automatic interaction continuation with `automatic_continuation`, `continuation_token`, `requires_continuation`, `continuation_ids`, and a video upscale task parameter.

└──▷ GET THIS VERSION ▾

npm

```
$ npm install @google/genai@2.32.0
```

git

```
$ git clone --branch v2.32.0 https://github.com/googleapis/js-genai.git
# already have the repo? check out this version:
$ git checkout v2.32.0
```

named in the project's README · checked 30 Oct Wrong command?

Enable automatic continuation of long interactions by setting `automatic_continuation` in `interactions.create`, with `continuation_token` and `requires_continuation` status now available on responses.

Track which interactions are linked by reading the `continuation_ids` field now declared on the `Interaction` type.

Add a video upscale task parameter to video generation requests.

NEW SUPPORTED PLATFORM Video understanding in Gemini Developer API

WAS THIS USEFUL? 👍 👎

Google Gen AI Python SDK

Feedback ⓘ v2.34.0 NOTES ↗

The Google Gen AI Python SDK provides Python interfaces to Gemini model generation, streaming, tools, and embeddings.

Adds interaction continuation tokens, automatic continuation, a video upscale task parameter, video understanding in the Gemini Developer API, and per-call HTTP options for live connections.

└──▷ GET THIS VERSION ▾

pip

```
$ pip install google-genai==2.34.0
# already installed? upgrade to this version:
$ pip install --upgrade google-genai==2.34.0
```

git

```
$ git clone --branch v2.34.0 https://github.com/googleapis/python-genai.git
# already have the repo? check out this version:
$ git checkout v2.34.0
```

named in the project's README · checked 30 Oct Wrong command?

Resume long-running interactions automatically by passing `automatic_continuation` to `interactions.create`, and track paused sessions with the new `continuation_ids` field on `Interaction` and `continuation_token` field alongside `requires_continuation` status.

Pass a video upscale task parameter when generating videos.

Support `base_url`, `api_version`, and `headers` inside `LiveConnectConfig.http_options` to customise live connection requests per call.

Google Gen AI Python SDK now supports video understanding in the Gemini Developer API.

NEW SUPPORTED PLATFORM Python 3.15 support

RETIRED SUPPORTED PLATFORM Python 3.10 support dropped

└──▷ BREAKING ON UPGRADE

! Python 3.10 is no longer supported; upgrade to Python 3.11 or later before updating the SDK.

WAS THIS USEFUL?  

Hugging Face OpenEnv

Feedback ⓘ v0.11.0 NOTES ↗ RANK

An interface library for RL post training with environments.

> OpenEnv v0.11.0 adds a Kubernetes container provider, a Fraud Triage RL environment, and a local `openenv serve` CLI command, alongside smaller improvements to Docker volume handling, schema exposure, and deployment tooling.

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

01 **`--public` flag in HF deployment script** NEW 70

Added a `--public` flag to `prepare_hf_deployment.sh` to make a Hugging Face Space public, skipping no-op visibility updates.

— Exact script and flag given, directly runnable v0.11.0

02 **`openenv serve` CLI command** NEW 65

New `openenv serve` CLI command serves OpenEnv locally via uvicorn.

— Runnable command named with its mechanism v0.11.0

03 **`/schema` endpoint for reset schema** NEW 65

Reset schema is now published via a new `/schema` endpoint.

— Concrete endpoint a reader can query directly v0.11.0

THINNER COVERAGE BELOW

04 **Volume mount forwarding in LocalDockerProvider** IMPROVED 55

`LocalDockerProvider` now forwards volume mounts so container-bound data is accessible inside sandbox runs.

— Named class and behaviour, no config example v0.11.0

05 **Self-hosted endpoint for pelican_svg judge** IMPROVED 50

The `pelican_svg` judge can now use a self-hosted endpoint.

— Named component but no config specifics v0.11.0

06 **Kubernetes container provider** NEW 40

OpenEnv adds a new Kubernetes container provider, letting environments run on Kubernetes clusters.

— No setup or config detail given v0.11.0

07 **Fraud Triage RL environment** NEW 40

Adds a Fraud Triage environment for transaction risk-decisioning reinforcement-learning tasks.

— Names the environment but not its mechanics v0.11.0

08 **`posttrain_env` prototype (RFC 014)** NEW 40

Adds an experimental `posttrain_env` prototype, defined under RFC 014, for post-training workflows.

— Experimental, no usage detail provided v0.11.0

09 **Mode-aware MCP tools via JSON-RPC** NEW 35

Exposes mode-aware MCP tools via JSON-RPC.

— Bare description, no interface detail v0.11.0

WAS THIS USEFUL?  

siliconscope

Feedback ⓘ v4.8.0 NOTES ↗ RANK

Sudoless Apple Silicon system monitor (native SwiftUI GUI) with ANE / Media Engine / memory-bandwidth tracking

> SiliconScope v4.8.0 adds cross-account AI runtime detection and brings per-process GPU share visibility to Fleet on Linux.

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

01 **Cross-account AI runtime detection** NEW 57

SiliconScope now detects AI runtimes (Ollama , `oMLX` , `mlx_lm` , `vLLM` , `MTPLX` and others) running under a separate OS account, such as a dedicated `ai` user, and queries their APIs the same way it does for the current user.

— Names mechanism and runtimes but no UI/config steps shown v4.8.0

02 **GPU process share visibility in Fleet on Linux** NEW 54

Fleet on Linux now shows each GPU process's share of the card (SM %) and labels graphics processes such as display servers and desktop shells; requires Linux agent 1.5.3.

— Specifies metric and version requirement but no exact UI path v4.8.0

WAS THIS USEFUL? 👍 👎

sandboxd

Feedback ⓘ v0.3.23 NOTES ↗ RANK

Open-source, self-hosted AI app builder — an agent builds real apps in isolated sandboxes on your own server, each live at a preview URL. Self-host in one command.

> sandboxd adds configurable OpenAI- and Anthropic-compatible model connections for self-hosted gateways, a new Pi agent, a reworked prompt-first console, and MiniMax M3.1 Flash routing in OpenCode — alongside a breaking change to Claude Pro/Max authentication.

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

01 OpenAI- and Anthropic-compatible model connections NEW

85

sandboxd now supports configurable `OpenAI-compatible` and Anthropic-compatible model connections for local model servers and LiteLLM gateways. URLs, keys, model aliases and token limits are managed in Settings, then the agent, connection and model are selected per task.

Point sandboxd at a self-hosted LiteLLM gateway so air-gapped projects never call external APIs.

📍 In the console, go to Settings › Model connections, click Add connection, choose OpenAI-compatible, enter your LiteLLM gateway URL and API key, set the exact model alias and token limit, then select that connection and model when starting a task.

— Exact Settings path, config fields and a runnable setup example v0.3.23

02 Claude Pro/Max login removed on upgrade BREAKING

85

Upgrading removes any Claude Pro/Max login connected in Settings → Coding agents. To keep using a Claude subscription, open each project, choose Claude Code and click Sign in with Claude to authenticate per-project; Claude API keys are not affected.

— Gives exact migration steps a reader can follow v0.3.23

03 Prompt-first console rework IMPROVED

70

The console now centers on prompt-first app creation with stack and model selection, alongside a persistent coding conversation next to preview, code, terminal and Git views, responsive navigation, searchable commands, workspace versions, and grouped settings for model connections, agent accounts, Git credentials and security. It also adds guided credential dialogs, explicit secret rotation, draft preservation, unsaved-edit warnings, and explicit preview sharing and deployment limit displays.

— Names many UI areas but no exact navigation path v0.3.23

04 **MiniMax M3.1 Flash model in OpenCode picker** NEW 60

sandboxd adds MiniMax M3.1 Flash to the OpenCode model picker, routed through the direct MiniMax proxy; M2.7 is no longer shown in the ordinary picker.

— Names the model and proxy but no setup steps v0.3.23

THINNER COVERAGE BELOW

05 **Pi agent routing** NEW 40

A new Pi agent can be selected alongside a configured model connection and model per task, routing tasks through Pi.

— Names the agent but gives no mechanism or setup detail v0.3.23

└──▷ **BREAKING ON UPGRADE**

! A Claude Pro/Max login connected in Settings → Coding agents is removed on upgrade. To keep using a Claude subscription, open each project, choose Claude Code and click Sign in with Claude to authenticate per-project. Claude API keys are not affected.

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.41.0 NOTES ↗ RANK

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

> The OpenAI Python SDK's v3.41.0 release adds typed references and tool usage models to the Realtime and responses APIs, metadata filtering for agent sessions, and more reliable WebSocket reconnection during Realtime sessions.

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

01 **WebSocket reconnect recovers queued binary frames** IMPROVED

50

WebSocket connections now recover queued binary frames after a reconnect, preventing data loss during Realtime sessions.

– Explains mechanism and benefit but no code surface named v3.41.0

02 **Metadata filtering for agent sessions** NEW

35

Agent sessions can now be filtered by metadata when listing sessions.

– States capability but no endpoint or parameter named v3.41.0

03 **Typed references in Realtime response inputs** IMPROVED

35

Realtime response inputs now use typed references instead of raw strings.

– Brief statement without naming the specific types or fields v3.41.0

04 **Response tool usage models in responses API** NEW

35

New model types were added to the responses API to access response tool usage models.

– Mentions new models but not specific type names v3.41.0

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

Niko1221 Strata

Feedback ⓘ v0.2.8 NOTES ↗ RANK

Strata is a local inference engine with one-click install for Windows and Linux, providing an OpenAI/Anthropic-compatible API on localhost for models like Qwen and DeepSeek.

> Strata's v0.2.8 release adds Intel Arc B-series pipeline parallelism, an automatic four-card Radeon profile with higher HIP batch throughput, and a DeepSeek prompt-speed flag, while quietly flipping the 0.2.7 NVIDIA routing speed-up back to opt-in.

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

01 Four-card Radeon profile and HIP batch size increase NEW

90

Strata now activates a four-card Radeon profile automatically for four R9700 cards — 32 request slots in 4 groups, layer split 13/26/39, and a 2048 prompt chunk; disable it with `"four_card_profile": false` in the config. HIP (AMD) engines now batch up to 16 rows at once instead of 8, enabling the throughput needed by this profile.

Disable the automatic four-card Radeon profile when you want to manage request slots and layer splits yourself.

```json
"four_card_profile": false
```

— Concrete config key, numbers, and mechanism all named. v0.2.8

02 Pipeline parallelism for Intel Arc B-series GPUs NEW

88

Enable pipeline parallelism across Intel Arc B-series cards with `--pipeline-windows 2`, which starts the next decode window on the first card while the second card checks the current one; disable it with `--pipeline-windows 0`.

— Flag, values, and exact mechanism all given. v0.2.8

03 RTX routing speed-up now opt-in BREAKING

83

The 0.2.7 routing speed-up for 12 GB NVIDIA cards is now off by default, restoring 0.2.6 routing behaviour unless re-enabled; turn it on via Models > Native engine settings > Speed in the UI, or with `strata launch --set STRATA_ROUTE_PRIOR=1.5`.

Recover throughput on a 12 GB NVIDIA card (RTX 3060 etc.) by re-enabling the 0.2.7 routing speed-up that is now opt-in.

```
$ strata launch --set STRATA_ROUTE_PRIOR=1.5
```

— Exact flag, UI path, and before/after behaviour given. v0.2.8

04 **DeepSeek prompt-speed flag for long prompts** NEW 73

Enable the opt-in `DS4_FAT_ROWS` optimisation on CUDA for long prompts, which gives the busiest experts one bigger matrix multiply, yielding +2% to +5% throughput on RTX A6000; off by default.

— Named flag and measured gain but no exact invocation shown. v0.2.8

05 **`strata doctor` warns on missing native_experts.txt** IMPROVED 65

The `strata doctor` command now warns when a model folder lacks `native_experts.txt` before launch, naming the real cause rather than blaming the CPU.

— Named command and file, but no example run shown. v0.2.8

06 **ModelScope install source flag fixed** IMPROVED 65

Run `strata install --source modelscope` to install from ModelScope; the flag is now correctly passed through.

— Exact runnable command given, but detail is thin. v0.2.8

THINNER COVERAGE BELOW

07 **Rollback to previous engine component** NEW 55

Roll back to the previous engine with `strata components rollback`.

— Runnable command but no further detail on scope. v0.2.8

08 **GPU memory reservation flag for DeepSeek** NEW 50

Strata now sends `--reserve-mib` to DeepSeek for the GPU memory for other apps setting.

— Named flag but no usage example or value given. v0.2.8

└──▷ **BREAKING ON UPGRADE**

! RTX cards with 12 GB or less: the 0.2.7 routing speed-up (`STRATA_ROUTE_PRIOR=1.5`) is off by default; the engine now routes as 0.2.6 did unless you explicitly enable it.

WAS THIS USEFUL? 👍 👎

Microsoft ONNX Runtime

Feedback ⓘ

plugin-ep-webgpu/v0.5.0 NOTES ↗

ONNX Runtime runs machine-learning models across hardware and operating systems with an open-source inference engine.

Adds `adapterIndex` and `enableMatmulFp32Accumulation` provider options, concurrent independent-Session execution on ORT 1.28.3+/1.30.1+/1.31+, and a static build configuration.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch plugin-ep-webgpu/v0.5.0
https://github.com/microsoft/onnxruntime.git
# already have the repo? check out this version:
$ git checkout plugin-ep-webgpu/v0.5.0
```

Select a physical Dawn adapter independently of the ORT WebGPU context identifier using the new `adapterIndex` provider option, separate from `deviceId`.

Enable FP32 accumulation for `MatMulNBits` and its fused variants with the opt-in `enableMatmulFp32Accumulation` provider option (default `false`).

ONNX Runtime WebGPU Plugin EP now supports concurrent independent-Session execution on ORT 1.28.3+, 1.30.1+, and 1.31+, with Session-owned command recordings and separate buffer-reuse tracking; older hosts fall back to serialized compatibility.

Add a statically linked WebGPU Plugin EP build configuration.

NEW SUPPORTED PLATFORM AMD GPU subgroup-matrix kernel support

NEW DETECTION Native GatedDeltaNet and VarlenCausalConvWithState kernels

NEW DETECTION Sparse attention operators

UPDATED DETECTION PagedAttention extended with non-causal and per-head normalization

NEW DETECTION Unquantized MoE and expanded QMoE support

NEW DETECTION Block-scaled FP8 QMoE, INT2 expert weights, and mixed-width QMoE

UPDATED DETECTION GatherBlockQuantized extended to FP8 and FP4

NEW DETECTION GroupNorm, SkipGroupNorm, and 3D ConvTranspose operators

NEW DETECTION Hyper-connection primitives

NEW DETECTION VarlenNGramHashMapping operator and Qwen4-Exp Engram support

WAS THIS USEFUL?  

LLM Compressor

Feedback ⓘ 0.15.0 NOTES ↗ RANK

LLM Compressor quantizes and compresses language models for deployment with vLLM and other inference engines.

> LLM Compressor 0.15.0 adds HIGGS mixed-precision quantization with ILP-based bitwidth allocation, QADModifier block-wise distillation without a separate teacher model, and REAP/MoNe expert pruning for MoE checkpoints, alongside MXFP6 quantization schemes, offline SpinQuant rotations, and an Intel XPU GPTQ kernel; it also removes legacy pruning modifiers and raises the minimum Python version to 3.11.

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

01 **HIGGS mixed-precision quantization allocation** NEW

90

Select optimal per-layer quantization schemes for a target average bitwidth using the new `get_higgs_config` function and `HiggsMSECollectorConverter`, composable with both the `oneshot` and `ModelFreePtq` entry points. PuLP is now available as an optional `higgs` extra dependency to enable HIGGS ILP-based bitwidth allocation.

— Names functions and entry points but no full runnable example 0.15.0

02 **Block-wise distillation via QADModifier** NEW

88

Run block-wise quantization-aware distillation after `QuantizationModifier` or GPTQModifier using the new QADModifier, which caches calibration inputs and unquantized subgraph outputs as the teacher without requiring a separate teacher model. Set `reobserve_weights=False` to keep preceding quantization parameters fixed.

— Clear mechanism and a concrete flag, no full code sample 0.15.0

03 **MoE expert pruning and collapsing** NEW

88

Adds MoE expert optimization: `ExpertPruner` prunes MoE checkpoints directly from safetensors, generates calibration-only saliency reports by setting `sparsity=0.0`, and stores per-expert router weights, token counts, and saliency data via `report_path` in JSON format; MoNe collapses low-saliency experts into constant novice vectors by writing `moe_sparsity_config` with `method='novice'` and repacking novice experts into the checkpoint while keeping the router at full width.

— Names class, flag and output format concretely Mechanism and config key named, no runnable snippet

0.15.0

04 **Offline SpinQuant rotations before quantization** NEW

78

Apply offline SpinQuant R1/R2 Hadamard rotations directly to safetensors before model-free quantization using the new `SpinQuantConverter`, and pass converter chains to `model_free_ptq`.

— Names converter and pipeline integration point 0.15.0

05 **Removal of legacy pruning modifiers** BREAKING 68

`SparseGPTModifier`, `WandaPruningModifier`, `ConstantPruningModifier`, `MagnitudePruningModifier`, the OBCQ compatibility shim, and shared sparsity utilities have been removed; any recipe or code referencing them will break on upgrade.

— Lists all removed surfaces but no migration path given 0.15.0

THINNER COVERAGE BELOW

06 **Automatic worker count in model_free_ptq** IMPROVED 53

Set `max_workers='auto'` (now the default) in `model_free_ptq` to let the pipeline select worker counts automatically.

— Named config value and default change, no deeper mechanism 0.15.0

07 **MTP layer loading via Transformers MtpModel** NEW 48

Load MTP layers through Transformers' `MtpModel`, with automatic fallback to unchanged MTP weights for unsupported architectures.

— Names the class and fallback behavior briefly 0.15.0

08 **Python 3.11 minimum supported version** BREAKING 45

Python 3.11 is now the minimum supported Python version; installations on Python 3.10 or earlier will no longer work.

— Clear version requirement, no further nuance 0.15.0

09 **Centralized fused module mappings** IMPROVED 41

Fused module mappings are now centralized in `FUSED_MODULE_MAPPINGS` for consistent module resolution across the pipeline.

— Internal refactor named but no user-facing action 0.15.0

10 **MXFP6A16 and MXFP6A8 quantization schemes** NEW 40

LLM Compressor now supports the `MXFP6A16` and `MXFP6A8` quantization schemes.

— Bare addition of two named schemes, no usage detail 0.15.0

11 **torchrun without init_dist now warns instead of erroring**

38

BREAKING

Running `torchrun` without `init_dist` now produces a warning instead of raising an error, changing observable error-handling behavior.

— Behavior change stated plainly, no scenario detail 0.15.0

---

12 **Sequential prefetch permanently enabled** BREAKING 37

`sequential_prefetch` is now permanently enabled and can no longer be disabled.

— States the change but not its performance impact 0.15.0

---

13 **Deprecation of preprocessing_func** DEPRECATED 37

`preprocessing_func` is deprecated; recipes or code using it should migrate to the documented replacement.

— Replacement is referenced but not named 0.15.0

---

14 **Fused GPTQ Triton kernel on Intel XPU** NEW 33

LLM Compressor now runs a fused GPTQ Triton kernel on Intel XPU.

— One-line hardware support note with no further detail 0.15.0

└──▷ **BREAKING ON UPGRADE**

! `SparseGPTModifier`, `WandaPruningModifier`, `ConstantPruningModifier`, `MagnitudePruningModifier`, the OBCQ compatibility shim, and shared sparsity utilities have been removed; any recipe or code referencing them will break on upgrade.

! Python 3.11 is now the minimum supported Python version; installations on Python 3.10 or earlier will no longer work.

! `sequential_prefetch` is now permanently enabled and can no longer be disabled.

! `preprocessing_func` is deprecated; recipes or code using it should migrate to the documented replacement.

! Running `torchrun` without `init_dist` now produces a warning instead of raising an error, changing observable error-handling behavior.

WAS THIS USEFUL?  

Experiential Labs Experiential

Feedback ⓘ

v0.7.223 NOTES ↗ RANK

Experiential is the open source, zero markup gateway for BYOK, self-hosted and 1000+ marketplace models. It learns from your traffic to cut costs, recommend better models, and train a specialized model you own.

> Experiential shipped automatic prefix caching for the Gemini Developer API (AI Studio), with a new preflight step in the automatic Google prefix planner.

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

01 Automatic prefix caching for Gemini Developer API NEW

57

Experiential now supports automatic prefix caching on the Gemini Developer API (AI Studio), including system-only prefix support in the automatic Google prefix planner. The planner also gained a `generateContentRequest` `-shaped` `countTokens` preflight to determine cacheable prefixes before sending requests.

— Names the API, planner and preflight shape but no usage steps v0.7.223

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

OpenWhispr

Feedback ⓘ v1.11.0 NOTES ↗

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

OpenWhispr adds shared team dictionaries for Business/Enterprise, a Skills system for recurring instructions, a workspace switcher in the sidebar, and GPU/CPU fallback for Whisper.

└──▷ GET THIS VERSION ▾

Homebrew

```
$ brew install --cask openwhispr
# already installed? upgrade to this version:
$ brew upgrade --cask openwhispr
```

git

```
$ git clone --branch v1.11.0
https://github.com/OpenWhispr/openwhispr.git
# already have the repo? check out this version:
$ git checkout v1.11.0
```

Homebrew lists this version · checked 30 Oct Wrong command?

Save recurring instructions in Integrations → Skills so the assistant picks a fitting skill automatically.

Switch workspaces and invite people from the sidebar; your account now lives under your avatar in the top bar.

Business and Enterprise members now get their team's shared dictionary applied in dictation and meeting notes, with personal words taking priority.

Whisper now checks whether CUDA or Vulkan started successfully and falls back to CPU transcription if not.

NEW INTEGRATION New AI model support: GPT-6.1 Sol, Gemini 3.8 Flash, Claude 5.5 on Bedrock, Gemma 4, Qwen3.8

WAS THIS USEFUL? 👍 👎

Open WebUI

Feedback ⓘ v0.12.5 NOTES ↗ RANK

Open WebUI is a self-hosted chat interface for local and remote models, with RAG over uploaded documents, tool calling, and multi-user access control.

> Open WebUI's v0.12.5 release expands admin-facing controls across RAG/document ingestion, analytics, and notes, adding a session-level analytics explorer, multi-page notes with live presence, admin API key management, and a set of new environment variables for fine-tuning embeddings, OCR, telemetry, and UI visibility.

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

01 RAG and document ingestion admin controls NEW 85

Admin Settings > Documents gains several new RAG knobs: `RAG_EMBEDDING_BUFFER_SIZE` controls how many chunks are held in memory at once when embedding large documents (default 1000 chunks per batch); `MISTRAL_OCR_PARAMS` (or the new Parameters field) selects a specific Mistral OCR model and passes extra request options for endpoints like Azure AI Foundry; `RAG_EMBEDDING_QUERY_PREFIX` and `RAG_EMBEDDING_CONTENT_PREFIX` are now configurable from this screen to supply prefixes some embedding models require; and `RAG_RELEVANCE_THRESHOLD` can be set to a negative value via the Relevance Threshold field for rerankers that return raw scores below zero.

Prevent high memory use when ingesting a very large document by lowering the chunk buffer — useful in memory-constrained deployments.

```env
RAG_EMBEDDING_BUFFER_SIZE=200
```

— Names every config key and field; buffer size has a worked example v0.12.5

02 Terminal startup response timeout NEW 70

`AIOHTTP_CLIENT_TIMEOUT_TERMINAL` overrides the new 60-second default Open WebUI waits for a terminal to respond with its skills, `AGENTS.md`, or uploads on startup.

Give a slow-starting terminal agent extra time to register its skills before Open WebUI treats the connection as timed out.

```env
AIOHTTP_CLIENT_TIMEOUT_TERMINAL=120
```

— Exact env var, default value, and a runnable example v0.12.5

03 **Exclude health-check URLs from OpenTelemetry metrics** NEW

70

Health-check and probe URLs can now be excluded from OpenTelemetry request metrics via `OTEL_PYTHON_FASTAPI_EXCLUDED_URLS` or `OTEL_PYTHON_EXCLUDED_URLS`, matching the existing behaviour already applied to traces.

Keep noisy Kubernetes liveness probes out of OpenTelemetry request metrics so dashboards reflect real traffic only.

```env
OTEL_PYTHON_FASTAPI_EXCLUDED_URLS=/health,/ready
```

— Named env vars plus a concrete usage example v0.12.5

04 **Hide chat-input controls via Visible controls** NEW

65

Selected tools, skills, filters, web search, image generation, and code interpreter controls in the chat input can now be hidden via Visible controls in Admin Settings > Interface or the `UI_VISIBILITY` environment variable.

— Names the env var and UI path, no example given v0.12.5

05 **Sessions tab in Admin Analytics** NEW

65

Admin Analytics gains a Sessions tab listing every chat with its tokens, responses, errors, timing, saved replies, and tool-call outcomes; it requires `ENABLE_ADMIN_CHAT_ACCESS` to be enabled.

— Describes fields shown and the gating flag, but no example v0.12.5

06 **Multi-page notes with live presence** NEW

65

Notes now support multiple pages and subpages that can be added, renamed, reordered, moved, downloaded, linked to, and deleted from page tabs, with search and model context scoped page by page. Live presence indicators show every other user who has a note open, and clicking a person jumps to the exact page and position they are working on.

— Rich behavioural detail but no config key or command v0.12.5

THINNER COVERAGE BELOW

07 **Admin API key rotation and deletion** NEW

45

Administrators can now rotate or delete any user's API key from the Edit User dialog; a rotated key is displayed once for copying.

— Gives UI path but no further mechanism or limits v0.12.5

---

08 **display_file tool extended to chat and knowledge base files**

IMPROVED

45

The `display_file` built-in tool now works for files attached in the chat or stored in a knowledge base, not only terminal files, letting models render them inline in a reply.

— Names the tool but gives no usage example v0.12.5

---

09 **Generated images saved to terminal working folder** NEW

40

Generated and edited images are now saved to a selected terminal's working folder so the model can continue working on them with the terminal's tools.

— States the behaviour but no config or command to act on v0.12.5

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

Mozilla AI Otari

Feedback ⓘ v0.34.0 NOTES ↗ RANK

Open-source, OpenAI-compatible LLM gateway you run yourself. One endpoint for 40+ providers, with virtual keys, budgets, and usage tracking.

> Mozilla AI Otari's v0.34.0 release adds at-rest encryption for session traces, a deduplicated `GET /api/v1/models` endpoint, model-based session filtering, and a consolidated Overview page in the web UI.

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

01 Deduplicated models endpoint BREAKING

90

The `GET /api/v1/models` endpoint now lists each model once instead of once per provider, with Otari automatically selecting the provider to use. This is a breaking change: callers that relied on per-provider model entries will receive a different response shape.

Retrieve the deduplicated model list after upgrading, verifying Otari now returns one entry per model regardless of backend provider.

```
$ curl -s https://<otari-host>/api/v1/models | jq '.[].id'
```

— Named endpoint, before/after behavior, and a runnable example v0.34.0

02 Trace encryption at rest BREAKING

75

Session trace content is now encrypted at rest using `OTARI_SECRET_KEY` or AWS KMS, keeping session data private. This is a breaking change: deployments must set the key or configure KMS, and existing unencrypted traces will not be readable without migration.

— Names the config key and KMS option plus migration note v0.34.0

THINNER COVERAGE BELOW

03 Overview page consolidates Usage, Activity, Sessions IMPROVED

45

The web UI now folds the Usage, Activity, and Sessions views into a single Overview page.

— Names consolidated UI areas but no exact navigation path v0.34.0

04 Model-based session filtering in traces view NEW

25

The traces view now supports filtering sessions by model.

— Thin one-line description with no mechanism or UI path `v0.34.0`

└──▷ **BREAKING ON UPGRADE**

! Session trace content is now encrypted; deployments must set `OTARI_SECRET_KEY` or configure AWS KMS — existing unencrypted traces will not be readable without migration.

! `GET /api/v1/models` now returns each model exactly once with Otari selecting the provider; callers that relied on per-provider model entries will receive a different response shape.

WAS THIS USEFUL? 👍 👎

sample-connector-for-bedrock

Feedback ⓘ v0.0.44

NOTES ↗ RANK

Seamlessly invoke Amazon Bedrock or your custom models, enabling a smooth experience with AWS GenAI services.

> This window's headline addition is Claude Code Web, a browser-based coding agent on Amazon Bedrock AgentCore, alongside new pass-through providers for native Anthropic Messages and OpenAI Chat Completions routes, a Lambda Web Functions compute type, new per-image/prompt-cache/agent billing mechanisms, and a string of breaking API, deployment, and credential-handling changes.

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

01 New billing mechanisms and quota overspend handling NEW

90

Adds `price_per_image` to the model config for painter, nova-canvas, and azure-openai-image providers so images stay unbilled until the key is set; adds `price_cache_read` and `price_cache_write` model config keys for bedrock-converse prompt-cache billing (per token, falling back to `price_in` when unset, with 1-hour cache writes billed at `price_cache_write`); bills bedrock-agent usage from the token usage recorded in the agent's trace, with knowledge-base calls using `bedrockModelArn` billed from an estimate of 4 characters per token across the question, answer, and cited references; and lets spend over the monthly quota draw from the account balance, which can go negative by at most the last call's amount, with recharges atomic, propagating to all instances within 60 seconds via cache, and adding to the stored balance.

Enable per-image billing for a nova-canvas model so that each generated image is recorded and charged at a set price.

```json
{
  "provider": "nova-canvas",
  "price_per_image": 0.04
}
```

Enable prompt-cache billing on a custom bedrock-converse model that uses prompt caching, setting separate read and write prices.

```json
{
  "provider": "bedrock-converse",
  "price_in": 0.000003,
  "price_cache_read": 0.00000015,
  "price_cache_write": 0.00000375
}
```

— Config keys, fallback behaviour, and estimation method all named Mechanism explained but no config key or command to act on

v0.0.44 v0.0.43

02 **Claude Code Web browser coding agent** NEW 90

Adds Claude Code in the browser at `/cc`, enabled via the new `EnableClaudeCodeWeb` CloudFormation parameter — each API key gets one microVM and one S3 workspace, with model calls routed back through `/v1/messages` against the user's key — and `EnableWebSearch` (default `true`), which creates an AgentCore Gateway with a Web Search target in supported regions billed at about USD 7 per 1,000 queries outside BRConnector quotas. The connector also sets `"skipWebFetchPreflight": true` in each user's `~/.claude/settings.json` automatically when the key is absent, and a later update bumps the underlying image to Claude Code 2.1.310 via the `ClaudeCodeWebImageBuildId` stack parameter, fixing `/mcp` and `/cost` commands to run and display output correctly.

Enable Claude Code Web for a team so each API key gets its own browser-based coding session backed by Amazon Bedrock AgentCore Runtime, with web search on.

```yaml
EnableClaudeCodeWeb: true
EnableWebSearch: true
```

— Named parameters and an enablement example across two releases v0.0.43 v0.0.44

03 **Preset model catalog migrations and additions** IMPROVED 85

SQL migrations in `src/scripts/migrations/` run at start-up on every compute type except `lambda-web`, adding missing preset models and binding them to group 1 without touching existing model names (apply manually to `lambda-web` with `psql $DATABASE_URL -f src/scripts/migrations/2026-10-20-preset-models.sql`, `2026-10-20b-claude-chat-presets.sql`, and `2026-10-22-retire-providers.sql` in that order). Adds presets for `claude-opus-5-5-chat`, `claude-sonnet-5-5-chat`, and `claude-haiku-5-5-chat` on bedrock-converse (used by NoDB mode and knowledge-base summaries), seven GPT-6 and GPT-5.6

presets on bedrock-runtime-openai, and, for new databases, six Claude presets plus seven OpenAI-compatible presets.

Manually apply the v0.0.43 migrations to an existing database (e.g. a lambda-web stack) after taking a snapshot, in the required order.

```
$ psql $DATABASE_URL -f src/scripts/migrations/2026-10-20-preset-models.sql
psql $DATABASE_URL -f src/scripts/migrations/2026-10-20b-claude-chat-presets.sql
psql $DATABASE_URL -f src/scripts/migrations/2026-10-22-retire-providers.sql
```

— Migration files and preset names given with runnable commands `v0.0.43` `v0.0.42`

---

04 **Pass-through providers for Anthropic Messages and OpenAI Chat Completions**

NEW 85

Adds a `bedrock-runtime-anthropic` provider that passes `POST /v1/messages` through to Bedrock's native `/anthropic/v1/messages` route, forwarding `system`, `tools`, `thinking`, `cache_control`, and `context_management` fields unchanged, with per-session region and credential pinning to keep prompt caching effective. Adds a `bedrock-runtime-openai` provider that passes OpenAI Chat Completions requests through to Bedrock's `/openai/v1/chat/completions` route, enabling OpenAI-format models such as Kimi K3 and GLM.

Point Claude Code at BRConnector's new Anthropic pass-through so `thinking`, `cache_control`, and other native fields reach Bedrock unchanged.

```
$ export ANTHROPIC_BASE_URL=https://your-endpoint
export ANTHROPIC_API_KEY=your-api-key
export ANTHROPIC_MODEL=your-model
```

— Routes, forwarded fields, and target models all named `v0.0.42`

---

05 **Lambda Web Functions compute type and stack recovery** NEW

80

Adds a `lambda-web` ComputeType to the CloudFormation template that runs BRConnector on Lambda Web Functions with native response streaming, billing by active CPU, and request timeouts up to 900 seconds. Trigger a `lambda-web` package rebuild and new Web Function revision by changing the new `WebFnBuildId` stack parameter, or by pointing `SourceRef` at a commit hash; recover a stuck `lambda-web`

stack rollback with `aws cloudformation continue-update-rollback --stack-name <stack name> --resources-to-skip WebFnPackage`.

Recover a stuck lambda-web stack rollback when the WebFnPackage resource cannot roll back on its own.

```
$ aws cloudformation continue-update-rollback --stack-name <stack name> --resources-to-skip WebFnPackage
```

— Describes behaviour but no setup command given Runnable AWS CLI command and exact stack parameters given

v0.0.42 v0.0.44

---

06 **Tightened API responses, limits, and model validation**

75

BREAKING

`/user/model/list` now returns only `name`, `provider`, `price_in`, `price_out`, `multiple`, and `group_name`; `/user/thread/list` now returns only the caller's own threads with `limit` capped at 100, the same cap also applied to `/admin/thread/list`; the Session-Id header is capped at 128 characters, returning HTTP 400 on every route when exceeded; API keys are now generated from a cryptographic random source; and a model string `name/<id>` that is not itself a configured model name now returns HTTP 400 for every key (including Feishu bot, knowledge-base, smart-router, and azure-openai model ids containing `/`), with the `model_id` request body field now ignored.

— Five named endpoints/fields with exact limits and codes v0.0.43

---

07 **ECS alternative template credential and access hardening**

75

BREAKING

The ECS stack admin key is now read from the SSM parameter `/brconnector/admin-key-<stack name>` (via `aws ssm get-parameter --name /brconnector/admin-key-<stack name> --query Parameter.Value --output text`); the `AdminApiKey` CloudFormation output is replaced by `AdminApiKeyParameter`. The ECS alternative template's ALB is restricted to CloudFront-only access, removing the `LoadBalancerURL` output — use `CloudFrontURL` to reach the service.

— Exact SSM path and CLI command to retrieve new key v0.0.44

---

08 **Removed aws-executor, urls-reader, and web-miner providers**

65

DEPRECATED

The `aws-executor`, `urls-reader`, and `web-miner` providers are removed; the upgrade migration deletes all models, key bindings, and group bindings for these

providers (history is kept), and requests to these models now return HTTP 400. Take a database snapshot before upgrading.

— Named providers and upgrade caution, no restore command given v0.0.43

09 **Configurable stream usage reporting for OpenAI-compatible models**

60

BREAKING

openai-compatible and azure-openai streams now send `stream_options` to the upstream by default; set `streamUsage: false` in the model config to disable this for upstreams or Azure API versions that reject it.

— Exact config key given but no worked example v0.0.44

10 **Anthropic-format errors on /v1/messages** IMPROVED 60

Errors on `/v1/messages` now return Anthropic-format JSON (`{"type":"error","error":{...}}`) with matching HTTP status codes — for example 401 `authentication_error` or 403 `permission_error` — so Anthropic SDK clients can handle them natively.

— Exact error shape and codes named, no action required v0.0.42

THINNER COVERAGE BELOW

11 **Removed shared API key endpoint** DEPRECATED 55

The `/open/get-shared-api-key` endpoint is removed; BRClient users now enter their own key, and a key named `__shared-api-key__` becomes an ordinary key.

— Named endpoint and migration path, no command needed v0.0.44

12 **EC2 boot log credential hardening** IMPROVED 55

Keeps keys and passwords out of the EC2 boot log; new stacks write the BRConnector start command to the root-only `/root/brconnector-run.sh` (mode 0700) instead.

— Exact file path and permission mode named v0.0.44

13 **Stricter validation of image content in messages** BREAKING 55

Content blocks whose `source` has `s3Location` now return HTTP 400 instead of being processed — send the content inline. Image URLs using unsupported schemes, which previously sent an empty image, now also return HTTP 400.

— Before/after behaviour named, remediation stated briefly v0.0.44

14 **17-region deployment expansion** IMPROVED 50

Expands `ec2`, `lambda`, and `ecs` deployments to 17 AWS regions, including ca-central-1, sa-east-1, and all eu-west/eu-central/ap-northeast/ap-southeast regions.

— Regions and compute types named, no setup step given `v0.0.44`

---

15 **Lambda@Edge function ARN version bump** `BREAKING` 50

The Lambda@Edge function publishes a new ARN (version `v2`); set `LambdaEdgeVersionArn` of your BRConnector stack to the new `LambdaEdgeVersionArn` output.

— Exact parameter to update, but thin on why it changed `v0.0.44`

└──▷ **BREAKING ON UPGRADE**

! openai-compatible and azure-openai streams now send `stream_options` to the upstream; set `streamUsage: false` in the model config if your upstream or Azure API version rejects it.

! The `/open/get-shared-api-key` endpoint is removed; BRClient users must enter their own key.

! The ECS alternative template's `AdminApiKey` CloudFormation output is replaced by `AdminApiKeyParameter`; a stack update issues a new admin key — read it from `/brconnector/admin-key-<stack name>`.

! The ECS alternative template's `LoadBalancerURL` output is removed and the ALB no longer accepts direct access; use `CloudFrontURL` instead.

! The Lambda@Edge function publishes a new ARN (version `v2`); set `LambdaEdgeVersionArn` of your BRConnector stack to the new `LambdaEdgeVersionArn` output.

! Image models stay unbilled until `price_per_image` is set in the model config; previously image generation was not tracked at all.

! bedrock-converse prompt-cache tokens are now billed; deployments using prompt caching will see new charges appear in recorded costs.

! `lambda` and `lambda-web` stacks created with `KeepEc2` set to `false` must be recreated when updating to the 0.0.44 `quick-build-brconnector.yaml` template.

! Content blocks whose `source` has `s3Location` now return HTTP 400 instead of being processed; send the content inline.

! Image URLs that use unsupported schemes (which previously sent an empty image) now return HTTP 400.

! The `aws-executor`, `urls-reader` and `web-miner` providers are removed; the upgrade migration deletes all models, key bindings and group bindings for these providers (history is kept). Requests to these models return HTTP 400.

! A model string `name/<id>` that is not itself a configured model name now returns HTTP 400; the `model_id` request body field is ignored.

! Claude 3.x presets and NoDB defaults are removed; existing models are left as they are.

! `lambda-web` stacks do not receive 0.0.43 on a stack update — the deployment package is built only at stack creation and migrations do not run at start-up. To run 0.0.43 on an existing `lambda-web` stack, deploy a new stack or manually apply `2026-10-20-preset-models.sql`, `2026-10-20b-claude-chat-presets.sql` and `2026-10-22-retire-providers.sql` from `src/scripts/migrations/` in file-name order after taking a snapshot.

! Take a database snapshot before upgrading: the upgrade deletes models of retired providers and adds preset models.

WAS THIS USEFUL?  

DATA

◆ Vector DBs & RAG

Pinecone

Feedback ⓘ changelog · Oct 30, 2026 CHANGELOG ↗ RANK

Pinecone is a managed vector database that stores and queries embeddings for AI applications.

> Pinecone shipped a PostgreSQL extension that brings vector search directly into Postgres, alongside a major Terraform Provider overhaul (v5.0.0/v5.1.0) covering a new index description style, integrated embedding, on-demand read capacity, and breaking changes to pod-based index handling.

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

01 **In-database vector search via pg_pinecone** NEW 70

Run vector search inside Postgres using the `pg_pinecone` open-source PostgreSQL extension, choosing the `native` engine to keep search local or the `remote` engine to offload a table's search to a Pinecone index.

— Names extension and both engine modes, but no install/setup steps

changelog-20261030-45b3c85e

02 **New index description style with migration in Terraform v5.0.0**

NEW 70

Terraform Provider v5.0.0 adds a new style for describing indexes using `dense vector`, `sparse vector`, and `full-text-search` fields, supporting document indexes, with the ability to migrate existing vector indexes to the new style without replacing them.

— Describes mechanism and migration path but no exact config syntax

changelog-20261030-45b3c85e

03 **Terraform v5.0.0 breaking changes to pod-based indexes**

70

BREAKING

In Terraform Provider v5.0.0, new pod-based indexes can no longer be created, and some configurations accepted by earlier versions are now rejected at plan time; applies that change read capacity or scale a pod-based index now wait for the change to finish before proceeding.

— Clear before/after behavior but no migration guidance given changelog-20261030-45b3c85e

THINNER COVERAGE BELOW

04 **Integrated embedding and on-demand read capacity in Terraform v5.1.0**

NEW

50

Pinecone Terraform Provider v5.1.0 adds integrated embedding for document indexes, letting Pinecone embed a field's text with a hosted model, and adds on-demand read capacity for indexes.

— Two thin additions named but no config keys or usage detail `changelog-20261030-45b3c85e`

└──▷ **BREAKING ON UPGRADE**

! Terraform Provider v5.0.0: new pod-based indexes can no longer be created, and some configurations accepted by earlier versions are now rejected at plan time.

! Terraform Provider v5.0.0: applies that change read capacity or scale a pod-based index now wait for the change to finish before proceeding.

WAS THIS USEFUL?  

stashbase

Feedback ⓘ v2.18.3 NOTES ↗ RANK

stashbase is an open-source, local-first writing workspace where you write with Claude Code and Codex, drawing on your own sources.

> stashbase's latest releases focus on resilience and workflow polish: indexing now recovers gracefully from permission-denied folders, Agent replies get per-block code copying, Codex gains persistent permission modes, and usage diagnostics expand alongside smaller UI refinements.

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

01 Resilient indexing for permission-denied folders IMPROVED

70

stashbase now continues indexing readable files when a child folder denies access, keeping the existing index and prepared text for the skipped folders rather than failing. A named warning with a Retry action surfaces in both keyword search and meaning search so the user can re-attempt indexing once access is restored.

– Names the two search modes and the Retry action, but no config key v2.18.3

02 Persistent Codex permission modes NEW

60

Codex approvals now offer a choice of session or persistent permission when the native request supports it, with permission-mode changes applying to the next turn in the same conversation.

– Explains mechanism and timing but no exact command or setting name v2.18.1

THINNER COVERAGE BELOW

03 Copy individual fenced code blocks in Agent replies NEW

55

Users can copy a single fenced code block from an Agent reply, with the copy action accessible on hover, keyboard focus, and touch, even when the block wraps across multiple lines.

– Clear interaction described but no exact UI path or shortcut v2.18.3

04 Expanded usage diagnostics NEW

55

Usage diagnostics now include bounded delivery checkpoints, observed Agent tool totals, and project read/write/edit outcomes, with genuine foreground interaction inside embedded HTML now counted as reading activity.

– Lists diagnostic components but no way to view or access them v2.18.3

05 **Agent selection inheritance for new projects** IMPROVED

35

New projects now inherit the most recently selected Agent, while existing projects retain their own independently chosen Agent.

— States the behaviour but no mechanism or configuration detail v2.18.1

06 **Full-pane HTML previews** IMPROVED

20

HTML previews now use the full document pane rather than a smaller area.

— Bare statement of change with no further detail v2.18.1

WAS THIS USEFUL?  

infino

Feedback ⓘ v0.11.18 NOTES ↗ RANK

Embedded retrieval library built on Parquet. Fast, efficient, and scalable.

> Infino's latest releases focus on query efficiency — full-text search memory use drops from 178 GB to 4–22 GB and term-index build time falls from 69 s to 6.5 s — alongside new value_counts column profiling, mid-query SQL cancellation, hidden companion `supertable` tables, and a run of scan, cache, and vector-search I/O optimizations; a compaction config key was also renamed, breaking existing configs.

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

01 Mid-query SQL cancellation via catalog.cancel NEW

80

Cancel a running SQL query mid-flight by calling the `cancel` method on the catalog, including while the query is still opening its tables — enabling timeout and interrupt patterns in long-running workloads.

— Named method on catalog, but no exact call example v0.11.14

02 Compaction config key renamed BREAKING

80

The compaction config key `compaction.max_memory_mb` is renamed to `max_merge_input_mb`; existing config files using the old name will need to be updated.

— Exact before/after config keys with clear migration step v0.11.14

03 value_counts method on supertable columns NEW

75

Read per-value counts for any column without writing a SQL statement using the new `value_counts` method on a supertable, e.g. docs.value_counts("category"), useful for quickly profiling categorical fields in a large table.

Get a frequency distribution of values in a column without writing a SQL query — useful for quickly profiling categorical fields in a large table.

```python
counts = docs.value_counts("category")
```

— Runnable example included and method clearly scoped v0.11.18

04 Full-text search efficiency improvements IMPROVED

70

Full-text search gets several efficiency improvements: queries now bound memory on reordered superfiles, cutting peak usage from 178 GB to 4–22 GB and making large-corpus FTS practical on memory-constrained hosts; tables can now omit their term index entirely, reducing storage and memory overhead for tables that don't use full-text search; search adopts a denser `V10` posting format and faster query kernels across every query shape, reducing latency for BM25 workloads; BM25 search now uses table-wide statistics on every search path for more accurate scoring; and full-text search can accept a bounded score shortfall in exchange for fewer index reads, lowering I/O on large corpora.

— Concrete before/after numbers but no user-facing control Four named FTS efficiency changes, all automatic

v0.11.18 v0.11.17 v0.11.13 v0.11.7 v0.11.6

05 **Parallel term-index build for faster ingest** IMPROVED

70

Term-index build now constructs large layers in key ranges on a thread pool, reducing build time from 69 s to 6.5 s and unlocking faster ingest for large tables.

— Strong before/after numbers but purely internal change v0.11.18

06 **Vector search ranking and placement improvements** IMPROVED

70

Vector search gains several accuracy and performance improvements: `vector_search` now ranks within exactly the rows a `WHERE` clause keeps rather than re-ranking a broader candidate set; the grid is now frozen on graph-served tables, removing mid-query split and recalibration and enforcing a per-cluster rerank budget for more predictable recall and cost; and drained vectors are now placed by their nearest fine centroid for improved recall.

— Named method and behaviour change, no runnable example given Mechanism and benefit described, no tunable surface named Mechanism named briefly, no metrics or config given

v0.11.18 v0.11.11 v0.11.6

07 **Scan and superfile read efficiency improvements** IMPROVED

65

Filtered `LIMIT` scans now start other partitions only on demand, reducing unnecessary partition reads on queries that satisfy their limit early. Scans of tables larger than their cache no longer download complete superfiles, instead fetching only the byte ranges needed, and queries on large tables now open superfiles incrementally by range and size rather than all at once, reducing peak memory and I/O. Cacheless superfile reads now use read-ahead batches sized by bytes for cold-cache throughput, cold SQL scans prefetch the next row groups while decoding the current one to overlap download and decode, and the reader cache now downloads files in 32 MiB chunks over 8 parallel streams for higher throughput.

– Six automatic I/O optimizations with no user-facing controls v0.11.18 v0.11.17 v0.11.11 v0.11.6

08 **Cross-statement decoded-column caching option** NEW

60

A new `keep decoded columns` option (off by default) keeps decoded columns available across multiple statements, avoiding redundant decoding work in multi-step query pipelines.

– Named option and default given but no exact config syntax v0.11.18

THINNER COVERAGE BELOW

09 **Configurable optimize-pass steps via OptimizeOptions** NEW

50

Configure which steps run during an optimize pass via `OptimizeOptions` to control compaction behaviour.

– Names the config struct but not its individual options v0.11.13

10 **Hidden companion tables (supertables)** NEW 50

Infino now supports hidden companion tables that a caller declares beside a primary table, enabling side data to live adjacent to a supertable without appearing in its schema.

– Concept described but no declaration syntax or API shown v0.11.11

11 **Generation-aware disk cache invalidation** IMPROVED 45

A table's disk cache now follows the table's generation in the catalog, so stale cache entries from prior generations are no longer served after a table is updated.

– Clear internal mechanism, nothing for users to act on v0.11.11

12 **Lighter per-column manifest statistics for wide tables** IMPROVED 40

Infino's manifest now uses lighter per-column statistics on wide tables, reducing overhead for schemas with many columns.

– Bare overhead-reduction claim with no numbers v0.11.7

13 **Concurrent ingestion writers per table** NEW 35

Ingestion appends now allow multiple concurrent writers to the same table.

– Single-sentence description, no limits or API named v0.11.6

└──▷ **BREAKING ON UPGRADE**

! The config key `compaction.max_memory_mb` is renamed to `max_merge_input_mb`; existing config files using the old name will need to be updated.

WAS THIS USEFUL?  

uteke

Feedback ⓘ v0.22.0 NOTES ↗ RANK

The Brain for Your AI — Local-first memory engine for AI agents. Store, recall, and search memories with semantic embeddings.

> uteke v0.22.0 adds author attribution and room linking for memories, a configurable secret-detection privacy filter across the write path, sturdier export/import with room-link portability, and a store-identity endpoint to stop commands from hitting the wrong store — alongside three breaking migration changes on upgrade.

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

01 Secret-detection privacy filter across writes NEW

100

Adds a secret-detection privacy filter configured via `[privacy]` in `uteke.toml`, with `mode` set to `redact` (default), `strict`, or `off` and regex patterns under `allowlist`, overridable at runtime with `UTEKE_PRIVACY_MODE`. `POST /remember` and `POST /room/remember` return `redacted: true` when a secret is masked and refuse strict-mode writes with `400 {"error": "secret_detected", "message", "kinds"}`; `POST /import` reports `redacted` and `rejected_secret`; `uteke remember` prints which kinds of secret were masked and its `--json` output gains a `redacted` field; MCP `uteke_remember` and `uteke_update` fail in strict mode with a tool error starting with `secret_detected:` naming the kinds. The filter also refuses or masks secrets in tags, metadata strings, and `source` on `remember`, `update`, `import`, `set_source`, and room consolidation; `update_memory_detailed` and `PUT /memory` return `redacted` showing what was masked.

Enable strict privacy enforcement so any write containing a detected secret is refused rather than silently redacted.

```toml
[privacy]
mode = "strict"
allowlist = ["DEPLOY_TOKEN_[A-Z0-9]{8}"]
```

— Names every config key, mode, endpoint, error shape and env var. v0.22.0

02 Link existing memories to rooms NEW

90

Link an existing memory to a room without copying it using `uteke room link <memory_id> <room> [--author]`, `POST /room/memory` (LAB tier, body `{ memory_id, room, author? }`), or the MCP tool `uteke_room_link_memory`; linking twice is not an error and returns `already_linked: true`.

Link a memory that already exists to a shared room so teammates can find it, without creating a duplicate.

```
$ uteke room link a3f9 sprint-planning --author alice
```

— Names CLI command, endpoint with body shape, and MCP tool. v0.22.0

03 **Author attribution on every write path** NEW 90

Record who wrote a memory with `--author` on `uteke remember` and the `author` field on `POST /remember` and MCP `uteke_remember` (free text, 1–128 characters); `author` is returned on recall, search, list and get, and survives export and import.

Attribute a memory to a specific agent or user so recall results show who stored it.

```
$ uteke remember 'Agreed on gRPC for the internal API' --author
'architect-agent' --json
```

— Names flag, endpoint, field, length constraint and persistence scope. v0.22.0

04 **Lifecycle/maintenance config now global-only** BREAKING 90

The `[lifecycle]`, `[dream]`, `[aging]`, and `[maintenance]` sections are now read from the global `uteke.toml` only; a project-local `.uteke/uteke.toml` copy of those sections is silently ignored with a warning. Move them to the global file, or set `UTEKE_TRUST_PROJECT_CONFIG=1` to keep the old behaviour.

— Exact config sections, file scope and override env var named. v0.22.0

05 **Export/import validation and room link portability** IMPROVED 85

`uteke import --full` now asserts the input must be a `uteke export --full` bundle and errors before storing anything if it is not. Room links also now travel with `uteke export` and `uteke import`: exported memories carry a `rooms` field, `import` re-links them to rooms that exist in the target store, and reports `room_links` (new links) and `rooms_missing` in the CLI, `POST /import`, and the JSON result.

— Names flags, new fields, and JSON result keys precisely. v0.22.0

06 **Store identity check via GET /store** NEW 85

Identify which store a running `uteke-serve` is serving via the new `GET /store` LAB endpoint, which returns `{"id": "<16 hex digits>"}` (a SHA-256 prefix of the store

path, never the path itself); the CLI now compares this id before routing a command to prevent writes landing in the wrong store when `UTEKE_HOME` points elsewhere.

— Endpoint, response shape and safety mechanism all specified. v0.22.0

07 **Embedding cache rekeyed on upgrade** BREAKING 65

The persistent embedding cache (`embed_cache.db`) is rekeyed on first open (`PRAGMA user_version` 1); all cached vectors are deleted and the first embeddings after the upgrade are recomputed.

— Names file, pragma value and consequence of upgrade. v0.22.0

08 **Store schema migration to v22** BREAKING 60

Stores migrate to schema v22 on first open to add the `author` field — back up a production store before upgrading.

— Names schema version and cause but no further mechanism. v0.22.0

└──▷ **BREAKING ON UPGRADE**

! The `[lifecycle]` , `[dream]` , `[aging]` , and `[maintenance]` sections are now read from the global `uteke.toml` only; a project-local `.uteke/uteke.toml` copy of those sections is silently ignored with a warning. Move them to the global file, or set `UTEKE_TRUST_PROJECT_CONFIG=1` to keep the old behaviour.

! Stores migrate to schema v22 on first open to add the `author` field — back up a production store before upgrading.

! The persistent embedding cache (`embed_cache.db`) is rekeyed on first open (`PRAGMA user_version` 1); all cached vectors are deleted and the first embeddings after the upgrade are recomputed.

WAS THIS USEFUL? 👍 👎

GrowGraph OntoCast

Feedback ⓘ v0.8.1 CODE ↗ RANK

OntoCast generates RDF knowledge graphs from documents by co-evolving domain ontologies and fact graphs with an agentic LLM pipeline.

> OntoCast v0.8.1 lets runs target named fact graphs, exposes new gate and connectivity metrics, adds a diverged-ontology merge endpoint, and removes two now-obsolete validator functions.

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

01 Diverged ontology merge via PUT /ontologies/{iri} IMPROVED

90

Call `PUT /ontologies/{iri}` (or ingest_ontology_ttl(..., as_new_version=True)) to merge a diverged ontology by deriving the new version from every head, instead of receiving a `409`.

— exact endpoint, method, alternative call, and before/after behavior v0.8.1

02 Named fact graphs via facts_graph_iri NEW

85

Name the graph a run's facts are written to with `AgentState.facts_graph_iri`; when unset it falls back to the document namespace.

Direct a run's extracted facts into a specific named graph rather than the default document namespace.

```python
state = AgentState(...,
facts_graph_iri='https://example.org/graphs/run-42')
```

— exact config key, fallback behavior, and runnable example v0.8.1

03 Run connectivity and gate metrics NEW

70

Retrieve four new run metrics — `facts_untyped_inferred`, `facts_links_removed`, `facts_components`, and `facts_isolated` — to see how many instances the gate typed instead of dropping, how many links its repairs removed, and how the served facts hang together.

— names four metrics but no retrieval mechanism shown v0.8.1

04 Removal of connectivity validator functions BREAKING

65

`validate_and_connect_content_unit` and `RDFGraphConnectivityValidator.make_graph_connected` are removed; any code calling either will break on upgrade.

— names removed functions but no migration path given v0.8.1

---

05 **Skip redundant document conversion on reuse** IMPROVED

60

A run whose `AgentState` already carries `docling_doc` now skips conversion and reuses the previously converted document for chunking and namespacing.

— names field but gives no usage example v0.8.1

└──▷ **BREAKING ON UPGRADE**

! `validate_and_connect_content_unit` and `RDFGraphConnectivityValidator.make_graph_connected` are removed; any code calling either will break on upgrade.

WAS THIS USEFUL? 👍 👎

1c-ai-connector

Feedback ⓘ v1.8.9 NOTES ↗ RANK

ИИкона — AI-коннектор для 1С:Предприятие 8.3: LLM-провайдеры, агентская петля (function calling), RAG, MCP-сервер, мониторинг ошибок с ИИ-анализом, генератор диаграмм

> This window's core theme is the new agentic function-calling (tool use) loop in Expert Chat, built out across nine releases with multi-database connections, per-model tool/loop limits, structured tool-call error reporting, paginated metadata retrieval, per-round cost estimation, dialogue/session logging, and new personal ChatGPT and GLM Coding Plan subscription connections.

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

01 **Function-calling tool use in Expert Chat** NEW 93

Enable tool use in Expert Chat by turning on Supports tools (function-calling) in the model card and then enabling Use tools (`Использовать инструменты`) in the chat form, giving the model access to four built-in tools: metadata tree, object structure, configuration info, and a read-only data query scoped to user permissions. Query a model with tool use via the new `ЗапросКМоделиСИнструментами` developer API method, or pass tool use to a background request using the new optional fourth parameter `ИспользоватьИнструменты` on `ЗапросКМоделиВФоне` .

— Names exact toggles and API methods with four built-in tools. v1.7.3

02 **Per-model tool selection and agent-loop limits on Tools tab**

IMPROVED 93

Control which built-in tools are available per model on the Tools tab; an enabled but empty selection disables tools entirely, and calling-code restrictions narrow the list further. Set four per-model agent-loop limits on the same tab — rounds with tools, tool calls, single result size, and total dialogue text size — where a value of zero keeps the defaults of 5 / 10 / 2000 / 24000 respectively. Tool-result size limits now account for the full JSON of the tool result, and dialog limits are checked against serialized `messages` before each round and the final round.

— Names exact tab, four limits, and their default values. v1.8.0 v1.8.1 v1.8.6

03 **Personal ChatGPT and GLM Coding Plan connections** NEW

93

Adds personal ChatGPT connections authenticated via Sign in with ChatGPT , selectable from the existing chat alongside API-based models, and personal GLM Coding Plan connections using an isolated supported-client profile (Codex), also selectable from the existing chat. Install or update these via the `subscription-adapter` directory (source and dependencies in the release archive); update by applying `IIkona-`

`1.8.4.cfe` with no data migration required and existing API connections preserved, verifiable via `SHA256SUMS.txt` and `VERIFICATION.json`. Breaking: shared cache is disabled for personal connections with no automatic retry on interrupted requests of unknown outcome; saved responses and tool continuations are isolated per connection and become inaccessible after logout or account switch; a connection code stored under one address is not resent after the address changes.

— Names exact files, directory, and migration/update steps. v1.8.4

---

04 **Structured tool-call failure reporting across agent and MCP**

IMPROVED 90

Results from `КИИ_КоннекторИИ.ЗапросКМоделиСИнструментами` and `КИИ_АгентАнализа.ВыполнитьАнализСИнструментами` now include an `ОшибкиИсточников` field, an array of structures with `database_id`, `database_name`, and a safe `Сообщение` string for each database that failed a tool call; the chat interface shows a separate source-named message for such failures regardless of what the model said, and responses with DSML service markup lacking real `tool_calls` are assigned an error status with no commands executed. MCP audit now marks `isError=true` calls as unsuccessful and preserves the original request code consistently across local calls, remote databases, and agent calls without duplicate error log entries; query execution failures are logged to the event log under `КИИ.СерверMCP.Диспетчер` (searchable by request code) and agent tool errors under `КИИ.АгентАнализа.ОшибкаИнструмента`. Both the calling client and a remote IIkona instance must be updated to `IIkona-1.8.9.cfe` for failure details to propagate correctly.

Check whether any connected databases failed to execute a tool call and surface the safe error message to the user, even when the model's reply gave no indication of a problem.

```1c
# We left this command out. It contains characters that look like ordinary ones but aren't, so it might not do what it shows. Check the vendor's own release notes before you run anything.
```

— Names exact fields, methods, and event-log categories with an example. v1.8.9 v1.8.5

---

05 **Paginated metadata and object-structure retrieval** IMPROVED

88

Retrieve large metadata lists in pages via `get_metadata_tree` using the new `offset` (from 0) and `limit` (1–200) parameters, with the response including `has_more` and `next_offset` fields to advance through results. The `get_object_structure` method now returns a complete metadata structure and supports retrieving fields in pages rather than all at once.

— Names exact parameters, ranges, and response fields. v1.8.1 v1.8.6

06 **Per-round cost estimation with configurable tariffs** NEW

88

Enable cost estimation per response on the Cache and Tariffs (`Кеш и тарифы`) tab of the model card by specifying currency and per-million-token prices for standard input, output, provider cache read, and provider cache write; estimates are stored separately from provider totals alongside a tariff snapshot. The estimate now appears under each chat response, covering the full agent loop including the final request; when statistics are incomplete or round currencies differ the line reads 'Оценка по тарифам: не рассчитана' instead of hiding, and a known zero cost displays as 0.

— Names exact tab, price fields, and fallback display text. v1.8.1 v1.8.3

07 **Agent session logging and dialogue journal** NEW

78

The request log now records the question and system instructions on the Question and context tab for all expert requests, filling previously empty fields; for tool-based requests it saves messages, tool calls, tool results, and the function catalogue offered to the model as a JSON object with `messages` and `tools` fields, written after personal-data checks (including error rounds) without storing HTTP headers or the model authorisation key. A new dialogue journal links requests, rounds, tool calls, responses, and errors together for each conversation.

— Names tab, JSON fields, and what's excluded from storage. v1.7.4 v1.8.0

08 **Multi-database connections in ИИкона** NEW

60

Connect multiple 1C databases in the central ИИкона, with the agent passing `database_id` on every call and passwords kept in the BSP secure store rather than sent to the model.

— Names database_id and secure store but no exact UI path. v1.8.0

THINNER COVERAGE BELOW

09 **Pre-send request context preview** NEW

45

Preview the prepared request text — including post-personal-data-check content — on the Before sending tab before it reaches the model.

— Names the tab but gives no further mechanism. v1.8.0

10 Migration tool for personal-adapter response archiving NEW

35

A migration tool lets you move old personal-adapter raw responses to a protected archive following a pre-reviewed plan, preserving receipts, ready responses, and tool continuations.

— Describes purpose only, no named command or path. v1.8.6

└──▷ BREAKING ON UPGRADE

! If using a remote MCP database, both the calling client side and the remote IIkona instance must be updated to `IIkona-1.8.9.cfe`; failure details are only propagated correctly when both sides run 1.8.9.

! Shared cache is disabled for personal subscription connections; on a request interrupted with unknown outcome, there is no automatic retry.

! Saved responses and tool continuations are isolated per personal connection and become inaccessible after logout or account switch.

! A connection code stored under one address is not sent to a new address after the address changes — the code must be saved again after any address change.

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.145.0 NOTES ↗ RANK

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

> Docling's latest release adds explicit resource cleanup for loaded documents and automatic page orientation detection for RapidOCR-based scans.

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

01 **Explicit resource cleanup via InputDocument.release()** NEW

73

Call InputDocument.release() to explicitly free resources held by a loaded input document, enabling tighter memory control in pipeline workflows.

— Names exact method; mechanism and purpose clear but no usage example. v2.145.0

THINNER COVERAGE BELOW

02 **Automatic page orientation detection in RapidOCR backend**

IMPROVED

45

Docling's RapidOCR backend now detects page orientation automatically before OCR, improving accuracy on rotated scans without manual pre-processing.

— Explains behaviour change but no config flag or command given. v2.145.0

WAS THIS USEFUL? 👍 👎

Tencent WeKnora

Feedback ⓘ v0.9.0 NOTES ↗ RANK

Open-source LLM knowledge platform: turn raw documents into a queryable RAG, an autonomous reasoning agent, and a self-maintaining Wiki.

> WeKnora's v0.9.0 release centers on a built-in multi-purpose assistant with multi-agent delegation and scheduled automation, alongside a standalone graph search API and sync improvements across Feishu, DingTalk, and HTML import.

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

01 **Built-in WeKnora assistant with side panel and management tools**

NEW 70

WeKnora adds a built-in assistant with an operation catalog scoped to knowledge API areas, accessible from any page with page context. Its side panel gained trial runs, session grants, page links, and MCP/model/skill management, and the assistant can now manage agents, skills, FAQ entries, and knowledge bases, find skills and MCP services on the web, delete and move documents in batches, and collapses long streaming timelines while animating active timeline icons.

– Many named assistant capabilities but no exact commands or endpoints v0.9.0

02 **Sync and import improvements for Feishu, DingTalk, and HTML**

IMPROVED 60

The Feishu/Lark sync now mirrors folder structure, parses blocks, and handles embedded objects; the DingTalk datasource optionally preserves the source directory structure (off by default); and HTML upload optionally indexes only the body text, aligned with the URL import path (off by default).

– Names each integration and behavior but no config keys given v0.9.0

THINNER COVERAGE BELOW

03 **Scheduled agent runs with notification inbox** NEW 55

WeKnora adds scheduled agent runs with an in-app notification inbox, letting automation tasks run on a timetable without manual triggers.

– Clear capability description but no concrete config or UI steps v0.9.0

04 **Standalone graph search API** NEW 55

WeKnora adds a standalone JSON graph search API for querying the knowledge graph directly.

— Named API and format but no exact endpoint path v0.9.0

05 **Durable multi-agent collaboration with live task views** NEW

50

WeKnora adds durable multi-agent collaboration so agents can delegate to child agents and track progress in real time via live task views.

— Explains mechanism but no UI path or command given v0.9.0

06 **RRF controls in search settings popover** NEW

45

WeKnora adds RRF controls in a tabbed settings popover in search.

— Names the control and UI location but no further mechanism v0.9.0

07 **Unified agent permission model** NEW

30

WeKnora adds a unified agent permission model with grants and run modes.

— Bare mention of grants and run modes without detail v0.9.0

08 **Human-in-the-loop prompt persistence via shared broker** NEW

30

WeKnora adds human-in-the-loop prompt persistence via a shared broker.

— Only a name and mechanism hint, no further detail v0.9.0

WAS THIS USEFUL?  

liliang-cn CortexDB

Feedback ⓘ v2.141.0 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 two releases centered on observability and trust: a queryable recall audit log with a `/recalls` UI, embedder health endpoints, structured logging and rotating call traces, plus new provenance labels distinguishing verified memories from unverified guesses and tighter access filtering on `find_nodes`.

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

01 **Recall audit log with /recalls page** NEW 95

Enable a recall audit log — including the queries themselves — with `CORTEXDB_RECALL_LOG=1`; browse it on the `/recalls` page or query it via `recall_log_*` tools, with retention controlled by `CORTEXDB_RECALL_LOG_DAYS` (default 30 days) and `CORTEXDB_RECALL_LOG_MAX_ROWS` (default 100,000 rows). The log can also now be queried by `trace_id` after calls join and forward the caller's `traceparent`, linking distributed traces end-to-end.

Turn on the recall audit log so every query is recorded and visible at `/recalls` — useful for auditing what an agent retrieved during a session.

```
$ CORTEXDB_RECALL_LOG=1 CORTEXDB_RECALL_LOG_DAYS=7
CORTEXDB_RECALL_LOG_MAX_ROWS=50000 cortexdb-grpc
```

— Names config keys, defaults, UI path, tool prefix and runnable example v2.140.0

v2.141.0

02 **Structured server logging and call tracing** NEW 90

Control server log output with `CORTEXDB_LOG_FORMAT` (`text` / `json`) and `CORTEXDB_LOG_LEVEL` (`debug`, `info`, `warn`, `error`), and capture full call traces to a rotating JSON-lines file via `CORTEXDB_TRACE` and `CORTEXDB_TRACE_FILE`, with rotation governed by `CORTEXDB_TRACE_MAX_MB`, `CORTEXDB_TRACE_KEEP`, and `CORTEXDB_TRACE_MAX_AGE_DAYS`.

Emit structured JSON logs and capture a full call trace to a rotating file to feed a SIEM or log aggregator.

```
$ CORTEXDB_LOG_FORMAT=json CORTEXDB_LOG_LEVEL=info
CORTEXDB_TRACE=1
CORTEXDB_TRACE_FILE=/var/log/cortexdb/trace.jsonl
CORTEXDB_TRACE_MAX_MB=200 CORTEXDB_TRACE_KEEP=10
CORTEXDB_TRACE_MAX_AGE_DAYS=14 cortexdb-grpc
```

— Every env var named with defaults and a runnable command v2.140.0

03 **Embedder health and metrics endpoints** NEW 81

Monitor embedder health via `/healthz` and `/metrics`, with the probe interval set by `CORTEXDB_EMBED_PROBE_SECONDS` (default every 60 seconds; set to `0` to disable).

Probe the embedder every 30 seconds so `/healthz` and `/metrics` reflect its state quickly — useful when running cortexdb-grpc behind a load balancer health check.

```
$ CORTEXDB_EMBED_PROBE_SECONDS=30 cortexdb-grpc
```

— Named endpoints and config key with default and example v2.140.0

THINNER COVERAGE BELOW

04 **Confined API keys restricted in find_nodes** NEW 55

Confined API keys now call `find_nodes`, filtered by the facade's hidden-node filter, so restricted callers see only their permitted graph nodes.

— Names the endpoint and filter mechanism, no example given v2.141.0

05 **Recall self-explanation and degradation reporting** IMPROVED 45

Recalls now explain themselves and report when they were degraded, with a request ID carried from MCP through gRPC to the trace.

— Describes behavior but names no field, flag or query syntax v2.140.0

06 **Verified vs unverified memory labels** NEW 40

Memories now label whether they were verified or are an unverified guess, and recall keeps the two apart in results.

— Explains the behavior but no field name or query syntax given v2.140.0

WAS THIS USEFUL? 👍 👎

aegis-stack

Feedback ⓘ v0.17.0 NOTES ↗ RANK

A production-ready FastAPI platform with modular components and a built-in control plane.

> Aegis-stack's biggest window adds a Google Meet bot and live in-call AI assistant (Illiana), pull-request review gating via `patterns check`, RAG knowledge bases, full request replay on the Overseer map, a database browser with secret-column masking, and Gemini voice support — alongside several breaking changes to voice endpoints, service import paths, and dev infrastructure.

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

01 **Meetings service with Google Meet bot** NEW 95

Add a Google Meet bot service with `aegis add-service meetings` that joins calls as a named participant, keeps a live transcript, and writes structured notes (summary, decisions, action items, open questions, commitments, contradictions, positions) after the call ends; meeting data can then be sent, watched, searched and read from the `/meetings` page, the CLI, the API, and both Overseer UIs.

Add the Meetings service to an existing stack to get a Google Meet bot that transcribes calls and writes structured notes.

```
$ aegis add-service meetings
```

— Runnable command, named access surfaces, structured note fields v0.16.0

02 **Database browsing and secret masking in Overseer** NEW 90

Browse any table's rows, schema and relationships — with filters, sorting and export — in Overseer's new `Database > Tables` view, and run read-only SQL in the new `Database > Query` editor, with saved queries returning as named tabs; mark a model column secret with `info={"secret": True}` to suppress it in row views, the Query editor, and error and log output, and secret columns are never shown in `Database > Tables`; on PostgreSQL the query runs as a reader role that cannot see secret columns and refuses functions or views that could read around that.

Prevent a sensitive column (e.g. an API key stored on a model) from appearing in Overseer table views, the Query editor, and error logs.

```python
api_key: str = Field(..., info={"secret": True})
```

— Exact UI paths named, no API endpoint given Exact field syntax and example, clear enforcement surfaces

v0.16.0

---

03 **Pull request review and gating via patterns check** NEW

80

Run `patterns check` to get Illiana's verdict on a pull request: new breaches of gated rules block the change while known debt stands, and she posts reviews as her own GitHub App when one is configured.

— Runnable command with gating behaviour and GitHub App integration v0.17.0

---

04 **Plugin system extensions: tools, packaging, and Overseer pages**

NEW

80

Plugins can grant their own tools, declare a queue concurrency, and add system packages and Dockerfile build steps to the image, with new plugins scaffolded with a release workflow; plugins can also register pages in the htmx Overseer via `PluginWiring.overseer_pages`, including a rail entry, sections, and templates, mounted alongside built-in pages.

— Named plugin API surface across two releases, no example given v0.17.0 v0.16.1

---

05 **OpenAI speech model deprecation** BREAKING

80

The default OpenAI speech model changes to `gpt-realtime-2.1-mini`; `gpt-4o-mini-tts`, `tts-1`, and `tts-1-hd` are deprecated and will be shut down by OpenAI on Jan 6, 2027, and the new model does not support the `fable`, `nova`, or `onyx` voices.

— Exact model names and shutdown date, no migration command v0.17.0

---

06 **Service import path consolidation** BREAKING

75

`AIService` and `FinanceService` must now be imported from `app.services.<x>.service`; code outside a service may no longer import them from their previous paths. Service code also moved under `domains/`: imports of a service's internal modules must now come from its `service.py` instead.

— Exact import paths named across two releases v0.17.0 v0.16.0

07 **RAG dependency moves to chromadb 1.x** BREAKING 75

`ai[rag]` moves to chromadb 1.x; the upgrade of an existing `data/chromadb` directory is one-way — keep a copy before upgrading if you may need to roll back.

— Exact version and path named, with rollback guidance v0.16.0

08 **aegis remove-service now drops service tables** BREAKING 70

`aegis remove-service` now drops the removed service's tables (after exporting them) and comments out its settings in `.env`; previously the tables stayed in place.

— Named command and before/after behaviour, no export format given v0.16.0

09 **Shared tabbed and tab_panel macros** NEW 65

Use shared `tabbed` and `tab_panel` macros in your own pages to build tabbed drawer views like the new meeting, person and series views.

— Named macro functions, no code example given v0.16.0

10 **Voice settings endpoint removed** BREAKING 65

`POST /voice/settings` is removed; use `POST /voice/use` to change the voice on the active profile.

— Exact endpoint replacement given, minimal mechanism v0.17.0

11 **Illiana live in-call meeting assistant** NEW 60

Invoke Illiana mid-call by saying Hey Illiana so she answers questions live during a meeting; her bot card shows her current work, and 'What's this about?' is available on any transcript line.

— UI phrase and button named, no API or config given v0.17.0

12 **RAG knowledge bases in Overseer** NEW 60

Fill Illiana's knowledge bases from Overseer by upload, by a server path, or by selecting app parts from the twin; indexing runs in the background and embedding is offloaded to the worker queue.

— Three ingestion paths named, no API endpoint given v0.17.0

13 **Request replay on the Overseer map** NEW 60

Replay any of the last 200 requests across the Overseer map hop by hop, down to the line of code, with Illiana drawing flows, cards, and path cards for each part she names.

— Concrete 200-request limit, no command surface given v0.17.0

14 **File-based theme definitions** NEW — 60

Define themes as individual JSON files alongside the built-in `aegis` and `steward` themes; the toggle offers any file placed in that directory.

— Named built-in themes and file format, directory path not given v0.17.0

15 **Sign-in cookie names changed** BREAKING — 60

Sign-in cookies are renamed to `<project>_session` and `<project>_refresh`, signing every user out once on upgrade.

— Exact cookie names and forced sign-out noted v0.16.0

THINNER COVERAGE BELOW

16 **Gemini voice support for voice profiles** NEW — 55

Gemini joins OpenAI as a second speech provider for voice profiles; voices are designed in three steps — pick, set delivery, preview — and each voice wears its provider's logo.

— Named provider and flow steps, no config key given v0.17.0

17 **Dev compose stack binds to localhost only** BREAKING — 55

The dev compose stack now binds every published port to `127.0.0.1`; anything that reached the dev stack from another machine stops working.

— Clear before/after but no config override given v0.16.0

18 **Redis family view in Overseer** NEW — 50

View a Redis family in its own terms on Overseer's Redis page, including a slow log that shows what each query touched.

— Named UI page, thin on mechanism v0.16.0

19 **Multipart S3 uploads for large files** IMPROVED — 45

The S3 store now sends files larger than 8 MB as multipart uploads.

— Concrete threshold but automatic, no user action v0.16.1

20 **Empty media queue removed from arq stacks** BREAKING — 35

The empty `media` queue is removed from arq stacks.

— Minimal detail beyond the name itself v0.17.0

└──▷ ALSO FROM THESE RELEASES

└──▷ BREAKING ON UPGRADE

! The default OpenAI speech model changes to `gpt-realtime-2.1-mini`; `gpt-4o-mini-tts`, `tts-1`, and `tts-1-hd` are deprecated and will be shut down by OpenAI on Jan 6, 2027, and the new model does not support the `fable`, `nova`, or `onyx` voices.

! `POST /voice/settings` is removed; use `POST /voice/use` to change the voice on the active profile.

! `AIService` and `FinanceService` must now be imported from `app.services.<x>.service`; code outside a service may no longer import them from their previous paths.

! The empty `media` queue is removed from arq stacks.

! Service code moved under `domains/`: imports of a service's internal modules must now come from its `service.py` instead.

! Sign-in cookies are renamed to `<project>_session` and `<project>_refresh`, signing every user out once on upgrade.

! The dev compose stack now binds every published port to `127.0.0.1`; anything that reached the dev stack from another machine stops working.

! `aegis remove-service` now drops the removed service's tables (after exporting them) and comments out its settings in `.env`; previously the tables stayed in place.

! `ai[rag]` moves to chromadb 1.x; the upgrade of an existing `data/chromadb` directory is one-way — keep a copy before upgrading if you may need to roll back.

WAS THIS USEFUL? 👍 👎

Crow Desktop

Feedback ⓘ v0.1.4 NOTES ↗ RANK

Crow: build your second brain from the documents on your computer. Downloads for macOS, Windows and Linux.

> Crow Desktop now lets Claude, Claude Code, ChatGPT and Codex connect directly to a user's Crow wiki memory via MCP, with one-click setup on macOS, Windows and Linux.

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

01 **One-click MCP connections for AI apps** NEW 79

A new Connect AI apps sidebar panel lets users connect Claude (desktop), Claude Code, ChatGPT (desktop), and Codex to Crow's wiki memory with one button each on macOS, Windows, and Linux. The apps reach Crow through MCP via llmwiki2's memory server, exposing `recall`, `remember`, and `read_note` tools, served by `crow mcp` behind a local token; Crow also launches itself in the background the first time a connected app needs its memory, even if Crow was closed when the app started.

Connect Claude or ChatGPT desktop to your Crow wiki so the app can look up notes and save what it learns.

📍 In Crow, open the sidebar and click Connect AI apps, then click the button next to Claude or ChatGPT. On macOS, Crow quits and relaunches the AI app automatically so it loads Crow. If connection fails, the window shows the command or config lines to paste manually.

— Names tools, command, and panel but no endpoint or exact flags v0.1.4

WAS THIS USEFUL? 👍 👎

synthadoc

Feedback ⓘ v1.5.0 NOTES ↗ RANK

Synthadoc: An open-source LLM knowledge compilation engine that turns raw documents into structured, local-first wikis. A transparent, human-readable alternative to traditional RAG, which can be self-managed and self-improved without the use of any tools.

> Synthadoc v1.5.0 turns on semantic search by default, adds Kimi as a provider, unifies rollback across all resolvers, and gives the knowledge graph a cluster overview for large wikis.

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

01 **Semantic search enabled by default** NEW 95

Enable semantic search by setting `vector = "auto"` in `config.toml`; new wikis now ship with this as the default, fusing BM25 keyword scores with local `BAAI/bge-small-en-v1.5` embeddings. Chinese, Japanese, and Korean pages are automatically routed to BM25 only to avoid degraded results.

Opt an existing wiki into semantic search after upgrading — change one key to activate BM25+vector fusion for English pages.

toml

```
# In your wiki's config.toml
vector = "auto"
```

— Named config key, embedding model, and language routing logic v1.5.0

02 **Kimi provider via DashScope** NEW 95

Add Kimi as a provider via DashScope streaming, authenticated with `KIMI_API_KEY` or `QWEN_API_KEY`; supported models are `kimi-k3` (1M-token context), `kimi-k2.6` (256K context), and `kimi-k2.7-code` (coding specialist). Set `thinking = "disabled"` in `config.toml` for `kimi-k3` to avoid timeouts on long contexts.

Configure Kimi's 1M-context flagship model as the default provider, with extended thinking disabled to avoid timeouts on long contexts.

toml

```
[agents]
default = { provider = "kimi", model = "kimi-k3", thinking =
"disabled" }
```

Authenticate with DashScope so the Kimi provider can reach the API — set the preferred key or its accepted alias.

```
$ export KIMI_API_KEY=<your-dashscope-key>
```

— Exact env vars, model names, and config flag with runnable examples v1.5.0

03 **Shared rollback across resolvers** IMPROVED 85

Rollback is now unified across all three resolvers — contradiction, orphan, and content-health — via a shared `_rollback.py` mechanism. tool_rollback_last_fix(slug), restore_page_snapshot(), tool_list_recent_fixes(), and parse_rollback_intent() are now available regardless of which resolver last touched a page.

— Named functions and module but no usage example given v1.5.0

04 **Knowledge graph cluster overview** NEW 70

The knowledge graph now opens on a cluster overview for wikis over 300 nodes, grouping pages into named Louvain clusters with a clickable list. It loads in ~6 seconds at 5,000 pages across both the web UI and the Obsidian plugin.

— Behaviour and scale described but no config or command v1.5.0

05 **Heading-invariant check on rewrites** NEW 65

A heading-invariant check (heading_mismatch_error()) now rejects any rewrite that adds, removes, or renames a heading, enforced across both the content-health resolver (`tool_propose_and_apply`) and the contradiction resolver.

— Named function and resolvers but no usage example v1.5.0

06 **Unified provider enum in config** IMPROVED 65

All provider names — including `anthropic` , `openai` , `gemini` , `groq` , `minimax` , `deepseek` , `kimi` , `qwen` , `ollama` , `claude-code` , and `opencode` — are now defined in a single Provider(StrEnum) in `synthadoc/config.py` , with `KNOWN_PROVIDERS` derived as a `frozenset[str]` .

— Exact class and file named, no usage shown v1.5.0

WAS THIS USEFUL? 👍 👎

FalkorDB

Feedback ⓘ v4.22.3 NOTES ↗ RANK

A super fast Graph Database uses GraphBLAS under the hood for its sparse adjacency matrix graph representation. Our goal is to provide the best Knowledge Graph for LLM (GraphRAG).

> FalkorDB v4.22.3 introduces a tunable write-lock polling mechanism to stop slow writes on one graph from freezing the whole server.

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

01 **Write-lock poll interval config** NEW 93

A new `WRITE_LOCK_POLL_INTERVAL` setting (in milliseconds, default `5`) controls how write queries wait for the graph lock; it can be set as a module load argument or changed at runtime with `GRAPH.CONFIG SET`, and setting it to `0` restores the previous single blocking-wait behavior. Write queries now release Redis's global lock while waiting for the graph write lock, so a slow read on one graph no longer stalls every other command on the server.

Lower the poll interval at runtime to reduce write-query latency spikes on a heavily read graph without restarting the server.

```
$ GRAPH.CONFIG SET WRITE_LOCK_POLL_INTERVAL 2
```

Set `WRITE_LOCK_POLL_INTERVAL` at startup so write queries yield Redis's global lock quickly in multi-graph workloads.

```text
loadmodule /path/to/falkordb.so WRITE_LOCK_POLL_INTERVAL 5
```

— Named config key, default, runtime command, and startup example all given v4.22.3

WAS THIS USEFUL? 👍 👎

vts

Feedback ⓘ vector-transport-service-v1.3.0 NOTES ↗ RANK

VTS is a tool designed for the transformation and transportation of vectors and unstructured data.

> VTS 1.3.0 adds a ChromaDB source connector, real-time Milvus CDC synchronization, and restored PostGIS geometry migration, alongside broader Milvus type support, Elasticsearch read performance work, and index/partition controls — with two breaking changes to Elasticsearch checkpoints and boolean metadata validation.

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

01 New migration source connectors NEW

90

Adds a new ChromaDB source connector that reads IDs, embeddings, documents, URIs, and metadata via the ChromaDB REST v2 API with tenant and database selection, supports Bearer, `X-Chroma-Token`, and Basic authentication, maps integer metadata to typed Milvus fields, and lets users size `batch_size` and `id_split_size` within server and memory limits; also restores the PostgreSQL CDC connector, which now supports migrating PostGIS `GEOMETRY` columns from PostgreSQL to Milvus.

— Names API version, auth methods, and config keys; no runnable example given Names the connector and type but gives no config or command detail

vector-transport-service-v1.3.0

02 Milvus CDC synchronization support NEW

90

A new WAL-based CDC source and a CDC sink write mode synchronize Milvus data in real time, covering insert and delete operations with source-position checkpointing, configurable upsert behavior, and backup/CDC overlap handling. It requires Milvus 2.6.21+ with the `DumpMessages` API; CDC sink parallelism must be 1 and AutoID targets require `allow_insert_auto_id = "true"`.

— Specifies exact version, API, and required config flags for setup

vector-transport-service-v1.3.0

03 Expanded Milvus type support and metadata preservation

IMPROVED

65

Milvus type support now covers `TIMESTAMPTZ`, `GEOMETRY`, `TEXT`, Int8 vectors, nullable vectors, and per-field timezone conversion, with improved preservation of AutoID, nullable attributes, partition-key metadata, and field descriptions during migration.

— Lists many named types but no config path to enable them

vector-transport-service-v1.3.0

04 **Elasticsearch read performance and parsing improvements**

IMPROVED

60

Elasticsearch reads now use shard-level parallel reads, a float-vector conversion fast path, timezone-aware date parsing, and index split controls for faster, more accurate migration from Elasticsearch sources.

— Names mechanisms but no specific flags or settings given

vector-transport-service-v1.3.0

---

05 **Elasticsearch checkpoint format breaking change**

BREAKING

60

Elasticsearch checkpoints and savepoints from the previous source split format cannot be restored with v1.3.0 — affected migration jobs must be restarted from the beginning, with target cleanup or deduplication planned as appropriate. The documented supported Elasticsearch range is now 7.10–8.x.

— Gives version range and migration remediation steps vector-transport-service-v1.3.0

THINNER COVERAGE BELOW

06 **Index creation and partitioning controls during migration**

NEW

55

New `create_index` and `partition_num` settings let users control index creation and partitioning during migration.

— Names exact config keys though usage context is brief vector-transport-service-v1.3.0

---

07 **Stricter boolean metadata validation**

BREAKING

30

Invalid metadata values mapped to BOOL now fail conversion rather than silently becoming false.

— States the behavior change but no field or config name vector-transport-service-v1.3.0

---

08 **Workload identity support for stage buckets** NEW

25

Workload identity support is now available for stage-bucket access.

— Bare mention with no mechanism or configuration detail vector-transport-service-v1.3.0

└──▷ **BREAKING ON UPGRADE**

! Elasticsearch checkpoints and savepoints from the previous source split format cannot be restored with v1.3.0 — affected migration jobs must be restarted from the beginning, with target cleanup or deduplication planned as appropriate. The documented supported Elasticsearch range is now 7.10–8.x.

! Invalid metadata values mapped to BOOL now fail conversion rather than silently becoming false (stricter boolean validation).

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ updated · Oct 30, 2026

NOTES ↗ RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's latest release adds AI-powered and in-pane search across runs and traces, a redesigned ABAC settings experience with CSV export, finer-grained org permissions, and several breaking changes to deployment management and deletion.

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

01 **Exclude-metadata filtering for thread search** NEW 85

Thread searches can now exclude threads matching given metadata via the `exclude_metadata` parameter on `POST /threads/search` and `POST /threads/count`, letting callers filter out e.g. automation threads when looking for user-facing ones.

Exclude automation threads (e.g. Open SWE sidebar threads) when searching for user-facing threads in a deployment.

```
$ curl -X POST https://api.langsmith.com/threads/search \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"exclude_metadata": [{"source": "automation"}]}'
```

— Named endpoints and parameter with a runnable curl example snapshot-20261030

02 **ABAC settings page tabs with CSV export and history** NEW 70

The ABAC settings page now includes Attributes, Policies, and History tabs with search, URL-persistent filters, and CSV export; the History tab lists policy create, update, delete, attach, and detach events for a chosen time range.

— Names tabs, filters, and event types but only a UI path snapshot-20261030

THINNER COVERAGE BELOW

03 **Engine permissions for custom roles** IMPROVED 55

Custom roles can now be granted Engine permissions in Settings > Roles, with all four Engine permissions shown in the role editor.

— Names exact settings path and permission count snapshot-20261030

04 **Traces section in Deployment Monitoring** NEW 50

Deployment Monitoring now includes a Traces section with trace count, latency, and error rate charts.

— Names metrics shown but no deeper mechanism snapshot-20261030

05 **Custom deployment headers moved to Connect dialog** BREAKING 50

The Settings dialog no longer shows custom deployment headers; they are now managed only in the Connect dialog and stored in your browser.

— Names both dialogs and storage change, no example snapshot-20261030

06 **Deployment deletion now immediate and confirmed by phrase** 50

BREAKING

Deleting a deployment now removes its database and metadata immediately (previously delayed), and requires typing `permanently delete` to confirm.

— Clear before/after but no code surface snapshot-20261030

07 **CSV export of org members and API keys** NEW 45

Organization members and API keys can now be exported to CSV from the Settings page, including all rows matching the current tab, search, and workspace filters.

— Clear UI path, limited mechanism detail snapshot-20261030

08 **Cancel reason on cancelled runs** NEW 35

Cancelled runs now expose a `cancel_reason` field so callers can access an explanation for why a run was cancelled.

— Names the field but no example or mechanism detail snapshot-20261030

└──▷ **BREAKING ON UPGRADE**

! The separate Revisions tab in Deployment is removed; bookmarks to tabs after Overview now open the tab to their right.

! The Settings dialog no longer shows custom deployment headers; they are now managed only in the Connect dialog and stored in your browser.

! Deleting a deployment now removes its database and metadata immediately (previously delayed), and requires typing `permanently delete` to confirm.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ arize-phoenix-client-v3.6.0 NOTES ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix's biggest change this window is authentication now being enabled by default, alongside new DECISION and AGENT annotation kinds rolled out across the client, CLI, and server, plus runtime fetching of model cost data.

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

01 **DECISION and AGENT annotation kinds** NEW 75

Arize Phoenix adds `DECISION` and `AGENT` as new annotation kinds, expanding the annotation taxonomy beyond existing kinds, across the client, CLI, and server. The CLI exposes this via `--annotator-kind DECISION` and `--annotator-kind AGENT` flags in the `px annotate` and `*-annotations delete` commands, usable for span, document, trace, session, and experiment run annotations.

— Names exact CLI flags and commands across client, CLI, and server.

arize-phoenix-client-v3.6.0 @arizeai/phoenix-cli@1.19.0 arize-phoenix-v21.0.0

02 **Authentication enabled by default** BREAKING 70

Authentication is now on by default without needing to set `PHOENIX_SECRET`, lowering the barrier to securing new deployments. Deployments that previously ran without `PHOENIX_SECRET` and without auth will require migration — see MIGRATION.md.

— Names the env var and migration doc, explains before/after. arize-phoenix-v21.0.0

03 **Chronological sorting for get_spans** IMPROVED 70

`get_spans` results can now be sorted by `start_time` to retrieve spans in chronological order.

— Names the exact function and sort field. arize-phoenix-client-v3.6.0

THINNER COVERAGE BELOW

04 **Runtime model cost manifest fetching** IMPROVED 50

Cost tracking now fetches the model cost manifest at runtime instead of relying solely on a bundled static file, keeping pricing data current without a release.

— Explains mechanism change but no config surface named. arize-phoenix-v21.0.0

05 **Prompt version creation timestamps via API** NEW 40

Prompt version creation timestamps are now accessible via the API, letting clients retrieve when a prompt version was created.

— No endpoint or field name specified. `arize-phoenix-client-v3.6.0` `arize-phoenix-v20.27.0`

06 **TypeSafe Jev decision models in evaluators** NEW 40

Python classification evaluators now support TypeSafe Jev decision models for type-safe evaluation outputs.

— Names the model type but no usage mechanism given. `arize-phoenix-v20.27.0`

07 **Trace and session filter expressions** NEW 35

New filter expressions are exposed on the client, allowing traces and sessions to be filtered programmatically.

— No concrete syntax or method name given. `arize-phoenix-client-v3.6.0`

08 **system_state database table** NEW 28

Arize Phoenix adds a `system_state` table to the database schema.

— Bare schema addition with no purpose or usage described. `arize-phoenix-v21.0.0`

└──▷ **BREAKING ON UPGRADE**

! Authentication is now enabled by default; deployments that previously ran without `PHOENIX_SECRET` and without auth will require migration — see MIGRATION.md.

WAS THIS USEFUL? 👍 👎

overmind

Feedback ⓘ v0.1.22 NOTES ↗ RANK

The platform for continuously improving AI agents.

> Overmind added dedicated inference serving and a Modal-based deployment path for trained models, alongside tighter Console visibility into coding-agent activity.

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

01 Dedicated inference serving via OpenAI-compatible API NEW

37

Overmind now supports dedicated inference, serving trained and frontier models through an OpenAI-compatible API.

— Names API compatibility but no endpoint or config details v0.1.22

02 Modal app deployment for fine-tuned model serving NEW

37

Overmind now exposes a dedicated Modal app for deploy, enabling self-hosters to serve fine-tuned models via Modal workers, deploying serving workers directly through the platform.

— Mentions Modal workers but no setup steps or config v0.1.22 v0.1.21

03 Coding-agent activity surfaced in Console IMPROVED

28

Coding-agent onboarding and activity are now connected to the Console, surfacing agent progress and dataset preparation status in one place.

— Describes integration but no UI path or specifics v0.1.22

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.72.0 NOTES ↗ RANK

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

> Langfuse v4.72.0 adds PagerDuty alerting, trace-level filter conditions for evaluations, and expands AI Gateway observability with a new analytics tab and per-key attribution.

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

01 **PagerDuty alerting integration** NEW 55

Send alerts, project notifications, and prompt changes to PagerDuty by configuring a routing key on the automation.

— Names routing key config but no exact UI path v4.72.0

02 **Trace-level filter conditions for evaluations** NEW 37

Add filter conditions at the trace level when setting up evaluations.

— Thin description, no filter options or fields named v4.72.0

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.133 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 latest release adds reasoning effort control for judge models in online evaluation rules and smooths out the annotator and Ollie chat UI.

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

01 **Reasoning effort selection for judge models** NEW 50

Online evaluation rules now let users pick the reasoning effort level when using OpenAI and OpenRouter models as judges.

— Names surfaces but no mechanism or UI path detail 2.2.133

02 **Simplified annotator view and persistent Ollie chat** IMPROVED 20

The annotator view card and preview are now simplified, and Ollie chat sessions continue across views rather than restarting.

— Vague UI change with no specifics on what simplified 2.2.133

WAS THIS USEFUL? 👍 👎

agentic-os

Feedback ⓘ v1.8.36 NOTES ↗ RANK

Governance framework for AI coding agents. It runs them through a five-step workflow (plan, build, review, test, ship) where no step counts as done without evidence.

> agentic-os added a design onboarding guide and two new scaffold templates to support brand and design-handoff workflows.

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

01 Design handoff onboarding guide and scaffold templates NEW

75

Adds a new core reference guide at `.agentcortex/docs/guides/bringing-your-own-design.md` explaining how to use readable design-tool references, local exports, or committed Markdown/ASCII wireframes with the existing design-first workflow, and distinguishing a source link from actual inspection, approval, and visual comparison. Also adds two scaffold templates: `.agentcortex/templates/design-reference.md` for design handoff records, and `.agentcortex/templates/custom-brand-design.SKILL.md` for a native `custom-brand-design` skill covering project-owned brand notes, a metadata mirror, optional phase-scoped capability registration, and nested personal preferences.

– Names exact file paths and skill but no usage walkthrough shown v1.8.36

WAS THIS USEFUL? 👍 👎

PromptLayer

Feedback ⓘ October 29, 2026 NOTES ↗

PromptLayer is a platform for managing, evaluating, and observing prompts and LLM applications.

PromptLayer adds Decision Models support with a dedicated question editor and structured answer review in the request details view.

NEW INTEGRATION Run decision-style models directly in PromptLayer, with dedicated tooling for defining questions and reviewing answers.

Define decision questions and their answer options in the new Decision Questions editor in the model parameters panel.

Review structured decision answers in the request details view.

NEW INTEGRATION Claude Haiku 5.5 model support

WAS THIS USEFUL? 👍 👎

nightingale

Feedback ⓘ v9.3.0 NOTES ↗ RANK

Open-source, self-hosted alerting for metrics, logs, and databases. Includes a built-in AI agent and MCP server.

> Nightingale v9.3.0 adds six new native notification channels with built-in templates, extends its AI assistant with CloudWatch Logs support, and polishes the unit picker and table UI.

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

01 Native notification channels for six services NEW

55

Nightingale adds native notification channels for Microsoft Teams, Twilio, Pushover, ntfy, Splunk On-Call and LINE, each shipping with built-in message templates.

— Names six channels but no config detail or endpoint v9.3.0

02 Unit picker custom units and description search IMPROVED

35

The unit picker now accepts custom units and supports searching units by description.

— Describes behaviour but no exact UI path or config v9.3.0

03 Inline icon buttons for table row actions IMPROVED

30

Table row actions now expand into inline icon buttons when the row has enough space.

— Clear UI behaviour change but no named surface v9.3.0

04 CloudWatch Logs preset questions in AI assistant NEW

25

Nightingale's AI assistant now includes preset questions for CloudWatch Logs.

— Bare statement, no detail on scope or usage v9.3.0

WAS THIS USEFUL? 👍 👎

traccia-py

Feedback ⓘ v0.1.33 NOTES ↗ RANK

OpenTelemetry-native SDK for AI agent observability, tracing, evaluation, debugging, governance, and runtime policy enforcement. Framework-agnostic and built for OpenAI Agents, LangGraph, CrewAI, and LLM applications in production.

> traccia-py's v0.1.33 release centers on a new `traccia eval` CI command with structured exit codes, pass-rate/cost/latency quality gates, provider key forwarding, and per-question gating, alongside status and cost-tracking improvements to evaluate() and a deprecation warning for implicit prompt versions.

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

01 Pass-rate, cost and latency quality gates in traccia eval

NEW 90

`traccia eval` adds a pass-rate quality gate via `--min-pass-rate` or `--max-drop` (e.g. `--max-drop 0.05` means five percentage points of regression), plus regression caps via `--max-task-cost-increase`, `--max-judge-cost-increase`, and `--max-latency-increase`, where a value like `0.2` means a 20 percent relative increase versus baseline.

Block a CI pipeline when pass rate drops more than 5 points or cost rises more than 20 percent versus the baseline prompt.

```
$ traccia eval experiments/support_agent.py \
  --output summary.json \
  --dataset support-golden \
  --scorer exact_match \
  --min-pass-rate 0.9 \
  --baseline-prompt support-reply \
  --max-drop 0.05 \
  --max-task-cost-increase 0.2
```

— names every gate flag with concrete value semantics v0.1.33

02 traccia eval CI command with structured exit codes NEW

88

`traccia eval SCRIPT --output PATH --dataset NAME` runs a task script in CI, saves an experiment, and writes `summary.json`. Exit codes 0 through 5 signal pass, quality gate miss, bad command or baseline, untrustworthy evidence, failed save, and summary write failure respectively.

Block a CI pipeline when pass rate drops more than 5 points or cost rises more than 20 percent versus the baseline prompt.

```
$ traccia eval experiments/support_agent.py \
  --output summary.json \
  --dataset support-golden \
  --scorer exact_match \
  --min-pass-rate 0.9 \
  --baseline-prompt support-reply \
  --max-drop 0.05 \
  --max-task-cost-increase 0.2
```

— full command syntax and exit-code mapping given `v0.1.33`

---

03 **Deprecation warning for implicit prompt version** DEPRECATED

73

Passing `prompt` without `prompt_version` or `prompt_label` still attaches the production label but now writes a deprecation warning; the next SDK release removes that default entirely. Use `--prompt` with `--prompt-version` or `--prompt-label` in `traccia eval`.

— gives exact flags and migration path, clear before/after `v0.1.33`

---

04 **Provider API key forwarding for LLM-judge scorers** NEW

70

`traccia eval` forwards provider API keys from the environment to LLM-judge scorers via `OPENAI_API_KEY`, `ANTHROPIC_API_KEY`, `GEMINI_API_KEY` or `GOOGLE_API_KEY`, `GROQ_API_KEY`, `TYPESAFE_API_KEY`, or `TRACCIA_PROVIDER_KEYS`; none of these values are written into `summary.json`.

— all env vars named but no direct usage example `v0.1.33`

---

05 **Scorer status and cost tracking in evaluate()** IMPROVED

66

evaluate() now preserves scorer `status` on each result, keeping `status: error` with no score for execution errors including remote platform scorers, and tracks task cost separately from judge cost. Task cost now includes OpenTelemetry attribute mappings and priced provider child spans on the SDK tracer provider.

— explains mechanism and field but no runnable example `v0.1.33`

THINNER COVERAGE BELOW

---

06 **Per-question gating for Jev scorers** NEW

48

Each Jev question in `traccia eval` now becomes its own independently gated check, named as `{scorer}:{question}`.

— names the gate format but lacks usage example v0.1.33

└──▷ BREAKING ON UPGRADE

! Passing `prompt` without `prompt_version` or `prompt_label` still attaches the production label but now writes a deprecation warning; the next SDK release removes that default entirely. Use `--prompt` with `--prompt-version` or `--prompt-label` in `traccia eval`.

WAS THIS USEFUL?  

Braintrust

Feedback ⓘ updated · Oct 30, 2026 NOTES ↗ RANK

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

> Braintrust shipped asynchronous SQL exports, a Terraform provider for infrastructure-as-code management, PagerDuty alerting, a visual SQL dashboard builder, and tracing for Cursor agent sessions.

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

01 Asynchronous SQL queries via bt sql --async NEW

80

Run SQL queries asynchronously via `bt sql --async` or the API, fetching results in pages so large exports aren't limited by a single synchronous request.

Kick off a large SQL export without hitting synchronous request timeouts, then page through results as they arrive.

```
$ bt sql --async "SELECT * FROM experiment_logs WHERE created_at
> '2026-09-01'"
```

— Runnable command and mechanism given, no page-size limit noted. snapshot-20261030

02 Terraform provider for infrastructure-as-code management NEW

60

A new Terraform provider, installable from the Terraform Registry, manages projects, permission groups, access control, AI provider secrets, environment variables, automations, prompts, and functions as code.

— Lists managed resources but no provider name or resource syntax. snapshot-20261030

THINNER COVERAGE BELOW

03 Visual builder for SQL-based dashboard charts NEW

50

Build dashboard charts from a SQL query using a visual builder that edits the same underlying query, with support for switching between query text and visual mode; available on Pro and Enterprise plans and requires data plane v2.1.1+.

— Describes toggle and plan gating but no exact navigation path. snapshot-20261030

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

NoLabs nono

Feedback ⓘ v0.81.0 NOTES ↗ CODE ↗ RANK

The nono runtime runs AI agents in isolated environments with policy-controlled filesystem, network, and credential access.

> nono v0.81.0 adds a policy-query command for Unix socket access, opt-in Linux namespace isolation, and Landlock V9 detection, alongside a new RTK state policy group, richer proxy deny logging, and a breaking change requiring fully qualified profile names.

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

01 **Unix socket access check via `nono why`** NEW 83

`nono why --unix-socket <path>` queries whether a sandboxed agent's connection to a pathname Unix socket (e.g. the Docker daemon socket) is permitted under the active profile, without making a real connection.

Check whether a sandboxed agent is allowed to connect to a pathname Unix socket such as the Docker daemon socket.

```
$ nono why --unix-socket /run/docker.sock
```

– Runnable command given, but no detail on output format. v0.81.0

THINNER COVERAGE BELOW

02 **Opt-in namespace isolation (NEP-0004)** NEW 50

Enable opt-in namespace isolation on Linux via NEP-0004 so sandboxed processes can be placed in their own namespaces.

– No flag or config key given for opting in. v0.81.0

03 **Fully qualified package profile names required** BREAKING 50

Package profile names must now be fully qualified; unqualified names are no longer accepted, making profile references unambiguous.

– Breaking change stated but no migration example. v0.81.0

04 **`denial_category` field in proxy deny logs** IMPROVED 45

Proxy deny log lines now include a `denial_category` field so operators can distinguish why a given network request was blocked.

— Names the field but not its possible values. v0.81.0

05 **`rtk_state` policy group for profiles** NEW 40

Add the `rtk_state` policy group to profiles to grant the permissions that RTK state requires.

— Names config group but not what RTK state needs it for. v0.81.0

06 **Landlock ABI V9 detection** IMPROVED 40

The sandbox now detects Landlock ABI up to V9 in the environment, while keeping actual enforcement at V6.

— Describes behaviour but gives no actionable control. v0.81.0

└──▷ **BREAKING ON UPGRADE**

! Package profile names must now be fully qualified; unqualified names are no longer accepted.

WAS THIS USEFUL? 👍 👎

containarium

Feedback ⓘ v0.122.0 NOTES ↗ RANK

Open-source agent runtime — SSH-native isolation, eBPF egress policy, Kubernetes + LXC backends, GPU passthrough, MCP-native CLI

> Containarium shipped run-user session management, mandatory backup encryption, and per-run model tracking across two releases this window.

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

01 **Run-user status check and sign-in terminal** NEW 75

Containarium adds `GetBoxRunUserStatus`, a new read-only pre-check API to check whether a run user is signed in before performing operations, and a new `terminal as_run_user` command to open a sign-in terminal as the run user.

— Names exact API and command but little mechanism beyond that. v0.122.0

02 **Per-run model tracking in BoxRun** NEW 65

Containarium now records and reports the model used per run: `StartBoxRun` accepts a per-run model parameter and `BoxRun` surfaces it in the result.

— Names both API surfaces but no usage example given. v0.121.0

THINNER COVERAGE BELOW

03 **Required encryption for backups** NEW 55

Backups can now require encryption by passing `require_encryption` to `create_backup`.

— Names the flag and function but no further detail on encryption method. v0.122.0

WAS THIS USEFUL? 👍 👎

cplt

Feedback ⓘ 2026.10.29-115649-c087e0b NOTES ↗ RANK

cplt is a kernel-level sandbox for AI coding agents, running Copilot CLI, Claude Code, OpenCode, Gemini CLI, and others with git and gh guards.

> cplt shipped a wave of kernel-level sandbox hardening — fail-closed enforcement for unenforceable deny paths, Landlock LSM + seccomp-BPF confinement for MCP broker child processes, and an opt-in network namespace with a proxy bridge — alongside secure-by-default policy tightening for new configs and smoother onboarding via `cplt config init` and a new `proxy.deny_dns` macOS DNS lockdown.

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

01 proxy.deny_dns option to block macOS DNS resolver NEW

86

cplt adds the opt-in `proxy.deny_dns` setting in `.cplt.toml` to close the macOS DNS resolver inside the sandbox, blocking an outbound exfiltration path via the system resolver.

Prevent the sandboxed agent from resolving DNS on macOS, closing an outbound exfiltration path via the system resolver.

```toml
proxy.deny_dns = true
```

– Exact config key, file, and example value given 2026.10.28-173629-aabbdd0

02 Automatic starter config generation on install NEW

75

cplt now creates a starter config file automatically for new users on first install. Fresh apt (Debian/Ubuntu) installs point new users to `cplt config init`, which generates a starter `~/.config/cplt/config.toml`.

– Gives the exact command and generated config path 2026.10.28-193606-cc59dce 2026.10.28-173629-aabbdd0

03 Default hardening for newly generated configs IMPROVED

73

New configurations generated by cplt now deny access to `agents/` paths and saved approvals, deny writes to `.git/config` on Linux (closing a persistence route previously open on the Landlock-only path), strip vendor OTLP authentication headers to

prevent credential leakage via telemetry, and have the outbound proxy forced on by default; existing configurations receive a nudge to enable the proxy.

— Names every denied path and header but gives no exact config keys

2026.10.28-180408-d70b8d3 2026.10.29-071519-d364d7c 2026.10.29-083654-79f116f 2026.10.29-085957-035752c

04 **Kernel-level sandboxing for MCP broker child processes** NEW 60

cplt extends kernel-level confinement to MCP broker server child processes spawned on Linux, sandboxing them with Landlock LSM and seccomp-BPF.

— Names exact kernel mechanisms but no user-facing control 2026.10.29-114500-2715aaa

THINNER COVERAGE BELOW

05 **Fail-closed enforcement for unenforceable deny paths** NEW 50

cplt's sandbox now refuses to start when a config specifies a `deny` path that the host OS cannot enforce, surfacing the gap at startup rather than silently leaving the path unprotected.

— Explains the fail-closed behavior but names no flag or command

2026.10.29-115649-c087e0b

06 **Per-call audit logging in MCP broker** NEW 50

cplt's MCP broker now logs one line per `tools/call` invocation, giving operators a per-call audit trail of agent tool use.

— Names the logged call type but not the log location or format 2026.10.29-085957-035752c

07 **Opt-in network namespace with proxy bridge** NEW 45

cplt can enable a network namespace with a proxy bridge on Linux via a new opt-in sandbox flag, isolating outbound traffic at the kernel level.

— Mentions an opt-in flag without naming it exactly 2026.10.29-083654-79f116f

WAS THIS USEFUL? 👍 👎

Backslash

Feedback ⓘ

How AI Agents Store Your Secrets in Session History - Backslash (2026-10-28)

BLOG ↗ RANK

Backslash detects and prioritizes reachable vulnerabilities in cloud-native application code and dependencies.

> Backslash launched Scout, a free endpoint assessment that discovers AI agents, skills, plugins and MCP servers on developer machines and flags where they store sensitive session data.

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

01 Backslash Scout endpoint scanning for AI agents NEW

59

Backslash Scout is a free assessment that discovers AI agents, skills, plugins, and MCP servers running on developer endpoints and shows where secrets and keys are stored. It highlights risk from session stores that can hold live credentials, including `~/.claude/projects/<project>/<session>.jsonl`, `~/.codex/sessions/.../rollout-*.jsonl`, `~/.codex/thread_history_*.sqlite`, and `~/.cursor/chats/<hash>/<session>/store.db`, which can be exposed when sessions are resumed via Claude Code's `--resume` / `--continue` flags or Codex's `resume` command.

Scan your local Claude Code session history for leaked keys and tokens before they are exfiltrated by a malicious skill.

```
$ gitleaks dir --redact "$HOME/.claude/projects"
```

— Names Scout and session-store paths but gives no direct Backslash command or UI step.

How AI Agents Store Your Secrets in Ses…

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Google Gemini API

Feedback ⓘ changelog · Oct 30, 2026

CHANGELOG ↗ RANK

Gemini API provides Gemini models for generating text, images, and other multimodal content in applications.

> Gemini API tightened control over request logging for the Generate Content API and introduced a new command-line interface for generating content and running agents.

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

01 Request storage controls for Generate Content API

BREAKING

86

On the paid tier, the Generate Content API (`batchGenerateContent`) now stores requests by default (`store=true`) for new projects, while existing projects keep their current setting. Developers can opt out per-request by setting `store=false` on individual calls, or disable logging globally via the Settings panel of the Logs and Datasets dashboard.

Opt out of default request storage on a single Generate Content API call to prevent sensitive prompts from being logged.

```
$ curl -X POST
'https://generativelanguage.googleapis.com/v1/models/gemini-
pro:batchGenerateContent' \
  -H 'Authorization: Bearer $API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"store": false, "requests": [{"contents": [{"parts":
[{"text": "Analyze this payload"}]}]}]}'
```

— Names exact field, endpoint, and UI path for opting out changelog-20261030-05bc3e3e

└──▷ **BREAKING ON UPGRADE**

! New projects using the Generate Content API (`batchGenerateContent`) on the paid tier now store requests by default (`store=true`); existing projects keep their current setting.

WAS THIS USEFUL? 👍 👎

OpenAI

Feedback ⓘ changelog · Oct 30, 2026 CHANGELOG ↗ RANK

OpenAI provides AI models and APIs for building applications that generate and reason with text, images, and audio.

> OpenAI shipped a new multimodal model, GPT-6.1 Terra, alongside several enhancements to the Evals API, and expanded free API credits for ChatGPT Pro and Business subscribers.

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

01 GPT-6.1 Terra model NEW

70

OpenAI added `gpt-6.1-terra`, a new model for visual reasoning, computer use, and natural writing, callable via the chat completions endpoint.

Route visual reasoning or computer-use tasks to GPT-6.1 Terra for fast, efficient responses.

```
$ curl https://api.openai.com/v1/chat/completions \
  -H 'Authorization: Bearer $OPENAI_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gpt-6.1-terra", "messages": [{"role": "user",
"content": "Describe what is on screen."}]}'
```

– Names model id and shows runnable curl example changelog-20261030-b7ef2fb3

02 Evals API request improvements IMPROVED

65

The Evals API gained the `input_json` field to pass structured data directly in requests, support for including images in individual question instructions, an increase in the per-request question limit to 400 (up from 200 in public beta), and caching of shared input across requests to receive prompt-cache discounts.

– Names field and limits but no usage example changelog-20261030-b7ef2fb3

THINNER COVERAGE BELOW

03 Free API credits for ChatGPT Pro and Business NEW

40

Eligible ChatGPT Pro and Business subscribers can now claim up to $500 per month in free API credits by linking an API organization.

– States benefit and amount, no setup steps changelog-20261030-b7ef2fb3

WAS THIS USEFUL? 👍 👎

Anthropic

Feedback ⓘ updated · Oct 30, 2026 NOTES ↗ RANK

The Claude Developer Platform is the API for building applications on Anthropic's Claude models that generate, analyze, and automate work.

> Anthropic shipped a new beta Usage and Cost Analytics API for the Claude Developer Platform and quadrupled the Messages API request size limit.

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

01 **Usage and Cost Analytics API (beta)** NEW 90

New endpoints `/v1/organizations/analytics/usage` and `/v1/organizations/analytics/cost` give access to token usage and cost data, enabled via the `analytics-usage-cost-2026-08-24` beta header. Results can be filtered and grouped by model, workspace, API key, user, or service account.

— Names exact endpoints, beta header, and filter dimensions snapshot-20261030

THINNER COVERAGE BELOW

02 **Larger request size limit for Messages and Token Counting APIs**

IMPROVED 50

The Messages API and Token Counting API now accept request sizes up to 128 MB, raised from the previous 32 MB limit.

— States before/after limits but no usage steps snapshot-20261030

WAS THIS USEFUL? 👍 👎

Writer

Feedback ⓘ announced · Oct 27, 2026 BLOG ↗ RANK

WRITER is the enterprise AI agent platform trusted by Fortune 500 companies, built to help teams execute and scale on-brand, compliant work.

> Writer's biggest addition this window is customer-specific invisible watermarking for Palmyra-generated text, rolling out alongside a skill security scanner, a built-in data loss prevention guardrail, Model Context Protocol support, and new governance and editing controls for Playbooks, dashboards, and microsites.

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

01 **Invisible watermarking for AI-generated text** NEW 72

Writer embeds invisible, machine-readable, customer-specific watermarks — a statistical signal tied to each organization's private key — into text generated by Palmyra models, paired with a detector that decodes it, letting organizations trace their own AI-generated content independently of the model provider. Available by December 2, it is enabled by default and pre-configured to meet EU AI Act requirements for EU customers, while non-EU customers can enable it at any time. Enterprises can tune watermark strength per use case to trade detection reliability against output quality, scope watermarking to specific content types (for example blog posts but not ad copy, or email campaigns but not product descriptions), and use multi-bit watermarking to embed custom metadata such as a campaign ID to track where content surfaces and how widely it spreads.

— Mechanism and scope well described but no exact toggle/API named

launch-20261027-f870dd75 launch-20261027-9916bcff

02 **Approval checkpoints and publishing controls in Playbooks**

NEW 65

Writer 's Playbooks now support built-in approval checkpoints that pause agentic workflows until a designated reviewer signs off, enabling human-in-the-loop governance for high-stakes workflows, plus new admin controls letting administrators decide who can create Playbooks and who can publish them to shared libraries.

— Clear mechanism and admin control scope described launch-20261027-f870dd75

03 **Model Context Protocol support** NEW 60

Writer adds Model Context Protocol (MCP) support, letting users run Playbooks, fetch work, and pull in Writer Meet notes from any MCP-compatible client using existing Writer seats and permissions.

— Names protocol and capabilities but no setup command launch-20261027-f870dd75

THINNER COVERAGE BELOW

04 **Live data and editing in dashboards and microsites** IMPROVED

55

Dashboards and microsites now pull live data from connectors and other data sources, support theme customization, and can be edited by selecting specific sections and instructing Writer Agent to make targeted updates staged in drafts before publishing.

— Describes editing workflow but no named UI path or API launch-20261027-f870dd75

05 **Data loss prevention guardrail** NEW

50

Writer adds a built-in data loss prevention guardrail that blocks sensitive information — including PII, financial data, and credentials — from reaching models and specified teams.

— Names data categories but no setup steps given launch-20261027-f870dd75

06 **Skill security scanner for malicious instructions** NEW

45

Writer adds a skill security scanner that reviews every skill for malicious instructions, with an option to automatically disable flagged skills.

— States behavior but no config surface or scope detail launch-20261027-f870dd75

07 **Mid-task steering for Writer Agent** IMPROVED

45

Writer Agent now accepts mid-task steering messages and queued follow-up requests that can be reordered, so the agent adjusts course without restarting.

— Explains behavior change but no concrete interface detail launch-20261027-f870dd75

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

FrontiersMind Lumma-fev-4B-v2 Feedback ⓘ

released · Oct 30, 2026 MODEL ↗

Lumma-fev-4B-v2 from FrontiersMind was released on 28 Oct 2026. Open weights on Hugging Face.

WAS THIS USEFUL? 👍 👎

DeepSeek V4.1 Flash

Feedback ⓘ price cut · Oct 29, 2026

MODEL ↗

DeepSeek V4.1 Flash is a multimodal model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4.1-flash): $0.14 → $0.045 in, $0.42 → $0.50 out per million tokens (-68%), first seen 29 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 4.6

Feedback ⓘ price cut · Oct 29, 2026 MODEL ↗

GLM 4.6 is a chat model from Z.ai.

Price cut on OpenRouter (z-ai/glm-4.6): $0.55 → $0.50 in, $2.20 → $2.00 out per million tokens (-9%), first seen 29 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.2

Feedback ⓘ price cut · Oct 29, 2026 MODEL ↗

GLM 5.2 is a chat model from Z.ai.

Price cut on OpenRouter (z-ai/glm-5.2): $0.0341 → $0.006 in, $7.00 → $7.00 out per million tokens (-82%), first seen 29 Oct 2026.

WAS THIS USEFUL? 👍 👎

IBM Granite 4.2 8B

Feedback ⓘ price cut · Oct 29, 2026

MODEL ↗

Granite 4.2 8B is a chat model from IBM.

Price cut on OpenRouter (ibm-granite/granite-4.2-8b): $0.06 → $0.06 in, $0.25 → $0.15 out per million tokens (-40%), first seen 29 Oct 2026.

WAS THIS USEFUL? 👍 👎

Moonshot AI Kimi K3

Feedback ⓘ

price increase · Oct 29, 2026 MODEL ↗

Kimi K3 is a multimodal model from Moonshot AI.

Price increase on OpenRouter (moonshotai/kimi-k3): $0.239 → $0.192 in, $10.80 → $15.00 out per million tokens (+39%), first seen 29 Oct 2026.

WAS THIS USEFUL? 👍 👎

NVIDIA Nemotron 3 Super

Feedback ⓘ

context change · Oct 29, 2026 MODEL ↗

Nemotron 3 Super is a chat model from NVIDIA.

Context window grew from 262K to 1049K tokens on OpenRouter (nvidia/nemotron-3-super-120b-a12b), first seen 29 Oct 2026.

WAS THIS USEFUL? 👍 👎

ConvAI Innovations Laya

Feedback ⓘ v0.4.17 NOTES ↗

RANK

Non-autoregressive System 1 decision engine. Typed choice, score and yes/no decisions over any text in a single forward pass, in 100+ languages, with a router that picks the right checkpoint per request.

> Laya's v0.4.17 release focuses on SDK parity and router infrastructure: Rust and .NET SDKs gain language-temperature and calibration controls, the Go router gets a full model lifecycle with SHA-256 integrity pinning, and a new C++ SDK foundation is laid down.

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

01 **Language temperature and calibration parity for Rust and .NET SDKs**

NEW 77

The Rust SDK now supports `lang_temperatures` and per-call `lang`, `max_len`, and `head_max_len` parameters, matching capabilities already available in other SDKs. The .NET SDK gains the same `lang_temperatures` and per-call `lang`, `max_len`, and `head_max_len` parameters, plus new support for calibration files and histogram binning.

— Names exact params and config features but no usage example v0.4.17

THINNER COVERAGE BELOW

02 **Go router model lifecycle management** NEW 57

Laya's Go router adds a model lifecycle with a factory, leases, LRU eviction, and single-flight request deduplication to manage which checkpoints are loaded and served.

— Mechanism named but no interface or config shown v0.4.17

03 **C++ SDK foundation (laya-cpp)** NEW 57

Laya adds a `laya-cpp` C++ SDK, including a CMake skeleton and CI lane, as the foundation for a native C++ library.

— Names build scaffolding but no usable API yet v0.4.17

04 **SHA-256 checkpoint integrity pinning in Go router** NEW 50

Laya's Go router adds per-checkpoint SHA-256 pins to verify model integrity at load time.

— Clear mechanism but no config or command shown v0.4.17

05 **Go SDK presets package sample** NEW 40

Laya's Go SDK adds a sample using the presets package to demonstrate router configuration.

— Names concrete package but thin description v0.4.17

WAS THIS USEFUL? 👍 👎

H2O.ai

Feedback ⓘ announced · Oct 28, 2026 BLOG ↗ RANK

H2O LLM Studio - a framework and no-code GUI for fine-tuning LLMs. Documentation: https://docs.h2o.ai/h2o-llmstudio/

> H2O.ai shipped two new products this window: H2OSuperRouter, a cost-aware LLM routing gateway with agent monitoring, ensembles and a decisions API, and the TimeH2O family of open-weights zero-shot time series foundation models installable via pip.

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

01 TimeH2O zero-shot time series foundation models NEW

95

Install the open-source inference package with `pip install h2ofm` (Apache-2.0) to run TimeH2O models without writing a training pipeline. TimeH2O-v1 is a 108M-parameter zero-shot time series foundation model available on Hugging Face under the H2O Model Non-Commercial License, ranking #1 of 32 on fev-bench with a 91.9% win rate and skill score of 50.8. TimeH2O-v1-tiny is a 5.8M-parameter univariate zero-shot model for compute-constrained environments, with a GIFT-Eval MASE of 0.687 and CRPS of 0.468.

– Exact install command, licenses, sizes and benchmark numbers given

launch-20261026-6f738475

02 Agent inventory and quality gating across providers NEW

70

Inventory and watch agents from their own traffic, from OpenTelemetry traces, or from Databricks, AWS Bedrock, Vertex AI, Azure AI Foundry, Snowflake, and Salesforce; agents that fail declared quality bars can be blocked from reaching production.

– Names six source platforms and the blocking behaviour launch-20261028-c02445fb

03 H2OSuperRouter routing gateway and router creation NEW

65

H2OSuperRouter is a single-container LLM routing gateway that speaks the OpenAI chat API and routes each request to the model most likely to answer it at the lowest cost. Teams can ship a pre-trained router covering eleven models in one step, or build a new router from their own models and traffic data.

– Mechanism and setup options named, no exact command shown launch-20261028-c02445fb

04 Request tracing to observability platforms NEW

65

Monitor every request — model chosen, cost, latency, and failure — with traces exportable to `OpenTelemetry`, Langfuse, Datadog, Prometheus, `LangSmith`, Arize, and MLflow.

— Lists seven named export targets, no config detail given launch-20261028-c02445fb

05 **Decisions API with calibrated probabilities** NEW 65

Access a decisions API compatible with OpenAI and OpenRouter that returns one-token answers with calibrated probabilities read from token log probabilities.

— Named API and compatibility, but no endpoint path launch-20261028-c02445fb

THINNER COVERAGE BELOW

06 **Coding agents routed through pooled credentials** NEW 55

Run coding agents (Claude Code and Codex) through the router on pooled credentials, with each session attributed to the individual user.

— Names both tools and attribution mechanism clearly launch-20261028-c02445fb

07 **Per-key, per-person and per-team cost controls** NEW 50

Apply cost controls including per-key, per-person, and per-team spending and request-rate limits, period budgets, and spend alerts.

— Lists control types but no config keys given launch-20261028-c02445fb

08 **Offline evaluation, playground and live A/B testing** NEW 45

Run offline evaluation, a playground, and live A/B and shadow experiments to compare routers and models on real traffic before changes reach users.

— Named testing modes, no interface details given launch-20261028-c02445fb

09 **Dataset building and router retraining** NEW 40

Build labeled datasets from live user traffic or uploaded data, use judge models to label them, and retrain the router on those datasets.

— Workflow described in prose, no named surface or command launch-20261028-c02445fb

10 **Multi-model answer ensembles** NEW 30

Run ensembles that combine answers from several models instead of picking one, raising accuracy beyond what the best single model reaches alone.

— States the benefit but no mechanism or config launch-20261028-c02445fb

WAS THIS USEFUL?  

Liquid AI

Feedback ⓘ announced · Oct 27, 2026 BLOG ↗ RANK

Liquid AI builds efficient Liquid Foundation Models (LFMs) for on-device, edge, and cloud AI with low latency, privacy, and hardware-aware deployment.

> Liquid AI launched the LFM3 family of multimodal foundation models, adding video understanding, agentic tool-use support, and broad on-device deployment across phones, laptops, and edge hardware.

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

01 **LFM3 model family architecture and context window** NEW

72

LFM3 is a new generation of Liquid Foundation Models built on a hybrid tri-operator architecture combining gated short convolutions, sliding-window attention, and global attention, with a 256K-token context window and pre-training on 22T tokens. The models are available via Hugging Face, the Liquid AI Playground, and the Liquid AI API.

— Names architecture, context length and training scale but no commands

launch-20261027-4bd98cf0

02 **On-device fine-tuning and deployment for LFM3** NEW

65

LFM3 models can be fine-tuned locally on a single GPU and run on-device across phones (Samsung Galaxy S26 Ultra, OnePlus 15), Raspberry Pi 5, Apple M4 Max, AMD Ryzen AI laptops, and the Qualcomm Hexagon NPU.

— Names concrete hardware targets and fine-tuning requirement, no exact commands

launch-20261027-4bd98cf0

THINNER COVERAGE BELOW

03 **Agentic harness support for LFM3-1.7B** NEW 56

LFM3-1.7B now supports agentic harnesses including Hermes Agent, Pi, and OpenClaw, with post-training across tool calling, web browsing, and enterprise workflows.

— Names specific harnesses and training scope, no setup steps launch-20261027-4bd98cf0

04 **Video input support in LFM3-VL-2B** NEW 45

LFM3-VL-2B now accepts video as input, making it the first LFM vision model to do so, enabling questions over recorded video from phones, lectures, or camera feeds.

— Describes the capability but no usage detail or endpoint launch-20261027-4bd98cf0

WAS THIS USEFUL?  

Alibaba Qwen Models

Feedback ⓘ qwen-audio-3.1-tts-next

NOTES ↗ RANK

Qwen is Alibaba's language and multimodal model family, including coding, vision, image, speech, and reasoning models.

> Qwen Models expanded its audio lineup with three new text-to-speech models: a unified multi-type audio generator, a voice-cloning TTS model with fine-grained emotion control, and a streaming TTS model for real-time voice interactions.

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

01 Voice cloning and emotion control in qwen-audio-3.1-tts-plus

NEW 65

`qwen-audio-3.1-tts-plus` is a text-to-speech model supporting multiple languages and dialects, streaming speech synthesis, and free-style instruction following. Emotion, tone, persona, speaking rate, and volume can be controlled via fine-grained tags, and voice cloning now handles noisy and reverberant reference audio more robustly, improving clarity and expressiveness.

— Detailed mechanism and before/after, but no command or flag given

qwen-audio-3.1-tts-plus

THINNER COVERAGE BELOW

02 Unified audio generation with qwen-audio-3.1-tts-next NEW

55

`qwen-audio-3.1-tts-next` is a unified audio generation model that combines speech, sound effects, and ambient audio in a single pass from text and timestamps. It supports multilingual TTS, single-speaker speech, multi-speaker dialogue, podcasts, cinematic soundscapes, ambient audio, and sound effects within one model.

— Describes capabilities and mechanism but no API call or config shown

qwen-audio-3.1-tts-next

03 Streaming synthesis with qwen-audio-3.1-tts-flash NEW

35

`qwen-audio-3.1-tts-flash` is a speech synthesis model with streaming output and a range of system voices, suited for voice assistants, real-time conversations, and customer service.

— Brief description of use cases, no technical specifics beyond that

qwen-audio-3.1-tts-flash

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 30, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue