NOW AVAILABLE The 512GB Mac Studio is on sale now, shipping February to March. See every configuration.

THE AI TOOLCHAIN

NO. 084

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 5, 2026 · EVERY DAY

EDITIONS **tail** | grep | head | ls | diff | uniq | perf

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 348 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 96 of 101 tools

▾ // THE BRIEF

HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: GPT-5.5 | Opus 5

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

vim ? help

Three unrelated things stand out today: Context Mode keeps almost all raw tool output out of agent context, CrewAI can benchmark model/role swaps against an existing workflow, and agents-cli makes Cloud Run agent identity and eval gates deploy-time concerns instead of conventions.

BUILD

Keep tool-heavy agent runs from flooding the prompt

Context Mode now routes command execution, batch search, URL indexing, and usage stats through MCP tools that summarize or store results outside the model window, keeping up to 99.4% of raw tool output out of context. That makes long research or coding sessions cheaper to run and less likely to lose the actual task under copied logs.

Context Mode

EVALUATE

Benchmark an agent workflow without rebuilding it as a separate eval harness

`crewai eval` runs model and role combinations against the workflow you already have and emits machine-readable briefs for coding harnesses. Teams can compare substitutions before rollout instead of manually cloning prompts, roles, and tasks into a one-off test script.

CrewAI

DEPLOY

Turn agent deployment checks into enforceable release gates

agents-cli added Cloud Run Agent Identity, eval pass/fail thresholds, pre-built Agent Runtime images, GKE dry-run validation, extension pinning, and project manifest tracking for app name and region. The deployment path can now fail on identity or quality conditions that were previously README instructions or post-deploy checks.

agents-cli

OPERATE

Recover the exact memory state an agent should have seen

uteke recall can now use `valid_from` and `valid_until`, with trust-tier and similarity filters, so an agent can retrieve memories as they existed at a specific time instead of only the latest version. That makes regression debugging and audit-style reconstruction possible without hand-exporting old session state.

uteke

Does **GPT-5.5** read better? 👍 👎

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.178 NOTES ↗ RANK

The xalgorix platform runs AI pentesting agents for reconnaissance, vulnerability detection, and exploitation workflows.

> xalgorix's latest release adds a new authentication-bypass detection check with its own report class, a JSON body field discovery tool, expanded body-injection options for SQL and template injection checks, and a fix for custom model gateway routing.

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

01 Authentication-bypass detection and reporting NEW

85

A new `verify_authbypass` check re-requests protected routes unauthenticated using equivalent path spellings (case fold, percent-encoding, trailing slash, dot and matrix separators), confirming a bypass only when the response status flips from denied to allowed and the body differs from both the denied response and the catch-all page. Confirmed bypasses are reported under the new `auth-bypass` report class (CWE-287/288/425), defaulting to medium severity and raised to high when the exposed body holds sensitive data.

– Full detection mechanism and reporting severity logic named, no invocation example. v4.6.178

02 JSON body field discovery tool NEW

78

The new `probe_body_contract` tool discovers which JSON body fields a POST/PUT/PATCH endpoint expects, classifying each field as required, recognized, rejected as unknown, or type-validated by reading the app's own validation errors. It is disabled in passive mode and records no finding in that mode.

– Named tool and classification behavior, no usage example given. v4.6.178

03 Body injection options for SQLi and SSTI checks IMPROVED

73

`verify_sqli` and `verify_ssti` gain `inject_in`, `body_type`, and `base_body` options to inject into JSON or form request bodies, sending sibling fields in `base_body` so the request still validates.

– Exact option names given but no concrete command shown. v4.6.178

THINNER COVERAGE BELOW

04 Verbatim model ID for custom gateway IMPROVED

50

When the provider is set to Custom Provider, a model ID such as `vendor/model` is now sent verbatim to the custom gateway endpoint, without stripping the prefix as a routing directive.

— Specific config behavior named but narrow scope. v4.6.178

05 **Expanded SQL error signatures for PostgreSQL and SQL Server**

IMPROVED 45

`verify_sqli` adds expanded engine-specific database error signatures to detect error-based SQL injection on PostgreSQL and SQL Server.

— Names engines and check but no mechanism beyond signature expansion. v4.6.178

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

ogcode

Feedback ⓘ v0.58.4 NOTES ↗ RANK

An open-source, local-first AI teammate and agent operating environment for turning intent into completed work—with tools, memory, permissions, and human oversight.

> ogcode's v0.58.4 release adds a benchmark harness with index caching and ships a cluster of context-window optimizations — lazy-loaded document tools, path-based memory file references, compaction fixes, and screenshot body capping — together trimming thousands of tokens and tens of kilobytes from prompts.

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

01 **Benchmark harness with index caching** NEW 90

Run the benchmark harness with `--build-index` to build a per-fixture index cache once and reuse it across attempts, then review solve rate, cost, wall-clock time, turns and tool calls in `benchmark/` using `ogcode-benchmark` 1.0.2.

Build the per-fixture index cache once before a benchmark run so each attempt restores it rather than rebuilding from scratch.

```
$ ogcode-benchmark --build-index
```

— Exact command, output location, and tool version given v0.58.4

02 **Lazy-loaded memory files in system prompt** IMPROVED 85

Memory files `MEMORY.md`, `AGENTS.md`, `AGENT.md` and the session memory file are now listed by path in the system prompt rather than inlined, and a pre-task check runs `file_map` followed by a ranged `read` or `grep` before the first edit, command or teammate spawn — dropping the assembled Build prompt by up to 60 KB on large projects.

— Names files, check sequence, and quantifies savings v0.58.4

03 **On-demand loading for document tools** IMPROVED 80

Document tools `pdf_index`, `read_pdf_page`, `docx_index`, `read_docx_page`, `query_sheet`, `latex_to_pdf` and `schedule` now load only when a step requires them via the `update_intent` matcher, cutting about 5.7 KB (roughly 1,400 tokens) of tool schema from build turns that need none of them.

— All tool names, matcher, and exact savings named v0.58.4

04 **Screenshot shedding with gateway body cap** IMPROVED

65

Screenshot shedding now enforces the gateway body cap with a 4 MiB window over the newest screenshot payload, replacing older tool frames with a short note while keeping the retained set stable across steps.

— Names 4 MiB limit and mechanism, no user action v0.58.4

05 **Context watermark sync in compaction** IMPROVED

60

Reactive and proactive compaction now settle the context watermark and pressure state before rebuilding the wire body, keeping the compaction-produced body as the body the next step sends and folding the summary into `messages[0]`.

— Mechanism explained but no direct user action v0.58.4

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.3035 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 shipped per-account near-cap request throttling, automatic relay balance-URL discovery, gateway-key-based routing views, remote-self provider detection, OpenCode 2 installation support, and a `--keep-data` uninstall flag, alongside smaller routing, gateway, and plugin fixes.

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

01 **Routing page enhancements** NEW 90

Routing now surfaces a missing-rule explanation with an example rule when a group has no routing rule configured; the Routing page adds a By key grouping option alongside the existing By request and By session views, with the selection persisted in `localStorage` under `magpie.routingGroupBy`, and each route now records the gateway key it arrived with as `CallerKey` (key ID) and `CallerKeyName` (key name at request time), exposed as `callerKey` and `callerKeyName` in the Route JSON — never the secret — so requests sent without a key carry neither field; and Search API and plan service requests are now listed on the Routing page, naming the API or service and the masked key or account, with their request and response kept in Recent calls and the request archive when archiving is enabled, and the API key masked out of all stored data.

— Thin description with no named surface or example shown. Names exact JSON fields and storage key for direct use. Describes behaviour and UI surfaces but no exact fields.

v0.1.3035 v0.1.3031 v0.1.3022

02 **Relay balance URL auto-discovery** NEW 85

Manually added relay providers now auto-probe for a balance endpoint using `/v1/usage` (sub2api) and `/api/usage/token` (new-api/one-api) when no Balance URL is configured, caching misses for 6 hours and unreachable relays for 5 minutes; the discovered query is re-checked after `balanceProbeHit` (24 hours) so a relay that has migrated to different software isn't read at a stale endpoint forever. The editor now shows where it found a relay's balance query, with a Use button to copy it into the balance fields, and the GUI shows the Balance URL field above the balance token field for custom providers, auto-opening the Advanced section when a Balance URL is already set.

— Named endpoints and config key, plus UI path for the Use button. v0.1.3030 v0.1.3029

03 **`--keep-data` uninstall flag** NEW 85

The `--uninstall` flag in `install.sh` now runs `magpie uninstall --yes --keep-data` first, cleanly disconnecting all agents without deleting settings or providers before removing the magpie binary; when magpie has already been deleted or is too old to run, `install.sh --uninstall` and `install.sh --purge` download the newest magpie to a temporary folder to perform this disconnect step.

Uninstall magpie but keep your settings, providers and usage data intact — useful when reinstalling or switching machines.

```
$ magpie uninstall --yes --keep-data
```

— Includes a runnable command and exact flag behaviour. v0.1.3023

04 **Package-manager install-script flags for agent binaries**

IMPROVED 80

magpie now passes per-package install-script flags — `--allow-scripts=<pkg>` on npm 12, `--trust` on bun, `--allow-build=<pkg>` on pnpm 10 — when installing agents whose npm package requires a postinstall script to place the platform binary, affecting `claude`, `cline`, `crush`, `deveco`, `mothx`, `omo`, `opencode`, and `snow`.

— Names exact flags per package manager and affected agents. v0.1.3017

05 **GLM Coding Plan web search via plan's MCP server** NEW

80

magpie now routes web searches for GLM Coding Plan (Zhipu/Z.ai) models through the plan's own Web Search MCP server at `ZhipuSearchPath` (`/api/mcp/web_search_prime/mcp`) on the plan's host, so searches spend the plan's allowance rather than a separate subscription's — active only when no searcher is named in Settings.

— Names exact MCP endpoint and activation condition. v0.1.3014

06 **Remote self-provider detection** NEW 75

magpie now detects a remote provider that points back at the local instance (a remote self) and excludes it from routing and model lists by default, persisting that determination across runs in `remote-self.json`; use Use it anyway (`KeepSelf`) to override and include it.

— Names persistence file and override flag for direct use. v0.1.3021

07 **Codex image detail mapping for relay APIs** IMPROVED 75

The gateway now maps Codex's `original` image detail value to `high` when forwarding requests to relay Responses APIs (such as OpenCode Go) that reject

`original`, while still sending `original` as-is to OpenAI's own API (`api.openai.com`, `chatgpt.com`), Azure OpenAI, DeepSeek's own API (`api.deepseek.com`), and remote magpie relays.

— Names exact endpoints and value mapping but no config to toggle. v0.1.3015

08 **Per-account near-cap request throttling** NEW 70

Accounts within `nearCapPoints` of their usage cap now take one request at a time, preventing multiple concurrent turns from pushing an account past its cap simultaneously; a turn that finds the account busy is routed to the next available account or waits its turn.

— Names the config key but not its default or exact units. v0.1.3035

09 **OpenCode 2 installation support** NEW 65

magpie can now install OpenCode as OpenCode 2 via `@opencode/cli` (npm) and `opencode.ai/v2/install`, keeping each installed line — OpenCode 1 via `opencode-ai`, OpenCode 2 via `@opencode/cli` — on its own upgrade track.

— Names install sources but no CLI command shown. v0.1.3017

THINNER COVERAGE BELOW

10 **Gateway routing fix for hand-made model groups** IMPROVED 55

magpie's gateway now correctly routes hand-made groups of embedding, rerank, speech, transcription, and video models when their providers have chat-model picks set; previously such groups returned a 404.

— Clear before/after but no config surface to act on. v0.1.3033

11 **Plugins page missing-agent notice** IMPROVED 45

The Plugins page now shows a clear message — including where magpie looks and the plugin's own notice — when a plugin's declared agent binary is not found on the local computer, instead of silently claiming the agent is added.

— Clear UI location but no named config or field. v0.1.3015

WAS THIS USEFUL? 👍 👎

ai-git-bot

Feedback ⓘ 1.30.0 NOTES ↗ RANK

The self-hosted AI workflow automation platform for Git repositories.

> ai-git-bot's 1.30.0 release adds first-boot setup with config export, Azure progress comments on pull requests, full raw payload storage for AI usage logs, and OpenRouter request attribution.

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

01 **Full raw payload storage for AI usage logs** NEW 60

AI usage logs can now store the full raw payload by setting the max length to `0`, which bypasses any truncation.

– Names exact config value but no field/file given 1.30.0

THINNER COVERAGE BELOW

02 **First-boot setup and config export** NEW 40

Adds first-boot setup, config export, and a throw-away trial instance, delivered via feat(config).

– Names feature but no concrete flags or steps 1.30.0

03 **Azure progress comments on pull requests** IMPROVED 25

The Azure integration now supports progress comments on pull requests.

– Brief description, no mechanism or config shown 1.30.0

04 **OpenRouter requests attributed to app** IMPROVED 25

OpenRouter inference requests are now attributed to the AI Git Bot app.

– States change but no detail on attribution mechanism 1.30.0

WAS THIS USEFUL? 👍 👎

keploy

Feedback ⓘ v3.6.217 NOTES ↗ RANK

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

> Keploy's v3.6.217 release focuses on test replay reliability, adding per-test browser seed determinism, parallel worker call isolation, in-process Docker Compose orchestration, and reduced false mismatches in repeated-read replay.

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

01 **In-process Docker Compose runner registration** NEW 40

Keploy adds an in-process compose runner that a build can register, enabling Docker Compose orchestration without shelling out.

– Names mechanism but no exact registration API given v3.6.217

02 **Per-test browser seed storage** IMPROVED 35

Keploy now stores each test's browser seed alongside its recorded start, so browser-based mock replay is deterministic per test.

– Explains mechanism but no flag or config name given v3.6.217

03 **Isolated parallel worker call tracking** IMPROVED 35

Keploy now keeps each parallel test worker's calls in its own test record, preventing cross-worker call contamination during concurrent runs.

– Describes behaviour but no concrete API or flag named v3.6.217

04 **Smarter repeated-read replay matching** IMPROVED 35

Keploy's proxy now replays a proven repeated read one additional time and skips one fewer excuse, reducing false mismatches in replay.

– Describes behaviour change but vague on exact mechanism v3.6.217

WAS THIS USEFUL? 👍 👎

kane-cli

Feedback ⓘ 0.10.2 NOTES ↗ RANK

Kane CLI by TestMu AI (Formerly LambdaTest)

> kane-cli 0.10.2 fixes TLS failures for Python steps behind inspecting proxies and sharpens failure reporting for locator and vision-based dropdown actions, while a breaking change removes redaction from sealed evidence packs.

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

01 **Certificate trust passed to Python steps** IMPROVED 80

Python steps now inherit kane-cli's certificate trust via the `SSL_CERT_FILE` environment variable, so runs behind a TLS-inspecting proxy or corporate network no longer fail with certificate errors.

— Names exact env var and failure scenario fixed 0.10.2

02 **Evidence packs sealed without redaction** BREAKING 60

Evidence packs are no longer redacted before sealing — secret values captured during a run will appear in the sealed pack. Users should treat evidence packs as sensitive artifacts and store them accordingly.

— Security-impacting change with clear handling guidance 0.10.2

THINNER COVERAGE BELOW

03 **Locator actions report specific failure reason** IMPROVED 47

Locator actions now report the reason for failure, distinguishing a missing element from a blocked click, without requiring log inspection.

— Describes behavior change but no named surface 0.10.2

04 **Vision dropdown steps detect failed selections** IMPROVED 43

Vision-based dropdown steps now report failure when a selection did not actually take, instead of claiming false success.

— Clear behavior fix, no configurable surface named 0.10.2

└──▷ **BREAKING ON UPGRADE**

! Evidence packs are no longer redacted before sealing — secret values captured during a run will appear in the sealed pack. Treat evidence packs as sensitive artifacts and store them accordingly.

WAS THIS USEFUL?

Letta Code

Feedback ⓘ v0.34.41 NOTES ↗ RANK

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

> Letta Code gained managed app port exposure for sandboxed applications and new subagent configuration options for runner selection and context inheritance.

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

01 **Subagent runner, type and context controls** NEW 65

New subagent configuration options: choose a runner for a subagent with `harness`, set the subagent kind with `subagent_type`, and pass the parent conversation history to external coding workers with `fork_context`.

– Names three concrete config fields with clear purpose v0.34.40

THINNER COVERAGE BELOW

02 **Managed app port exposure for sandbox agents** NEW 35

Sandbox agents can now expose managed app ports, allowing in-sandbox applications to be reached from outside the sandbox.

– Describes capability but no flag or command shown v0.34.41

WAS THIS USEFUL?

aerovato Operator Memory

Feedback ⓘ helper@v1.12.0

NOTES ↗ RANK

Operator Memory is a self-improving context engine for coding agents.

> Operator Memory's Claude Code harness now supports native auto-updates, cutting down on manual maintenance when updating the helper.

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

01 Native auto-updates for Claude Code harness IMPROVED

30

The Claude Code harness in Operator Memory now supports native auto-updates, reducing manual maintenance when updating the helper.

— States benefit but no mechanism, command, or config detail given. helper@v1.12.0

WAS THIS USEFUL? 👍 👎

mvschwarz OpenRig

Feedback ⓘ v0.7.8 NOTES ↗ RANK

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

> OpenRig's v0.7.8 release lets a designated person answer a runner seat's permission requests instead of the agent being auto-refused, adds a standalone conformance checker for new harness runners, and rounds out host management, mission-following and tmux-isolation commands alongside a breaking daemon and plugin upgrade.

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

01 Human-answerable permission requests for runner seats

95

BREAKING

Assign a person to answer a runner seat's permission requests with `rig seat set-permissions-to <seat> --to <person> --reason '…'`; use `--clear` to return to the default or `--deny` to opt into refusals. As a behavior change, runner seats following section 7 of `docs/reference/harness-runner-contract.md` now receive answers or permits instead of automatic refusals, so seats that should keep refusing must explicitly pass `--deny`.

Route a runner seat's permission requests to a specific colleague instead of the default person, so the right human is asked before the agent is permitted to act.

```
$ rig seat set-permissions-to my-runner-seat --to alice --reason
"Alice owns this project's infra decisions"
```

— Exact flags and the default behavior change are both given v0.7.8

02 Runner conformance checker NEW

90

Run a runner conformance check with `node docs/reference/runner-conformance/check-runner.mjs --runner '<runner command with {agent}>' --name <harness>`. It requires only Node and tmux — no daemon, model, credentials, or network — so new harness runners can be validated against the contract in isolation.

Validate a new harness runner against the answered-permission contract before connecting it to a live daemon.

```
$ node docs/reference/runner-conformance/check-runner.mjs --
runner 'dsh run --agent {agent}' --name dsh-harness
```

— Runnable command and exact dependency scope given v0.7.8

03 **Daemon migrations and bundled plugin upgrade** BREAKING

90

Restart the daemon after upgrading (`rig daemon stop && rig daemon start`); four database migrations (142–145) run on first start and cannot be skipped. The bundled `openrig-core` plugin advances from 0.1.23 to 0.1.24, with one line of its seat-continuity skill changed, and managed refresh now stores replaced files under `$OPENRIG_HOME/backups/managed-refresh/<id>/` — reapply any local edits you still need from that copy.

— Exact migration numbers, versions, backup path and restart command v0.7.8

04 **Isolated tmux server per OpenRig home** NEW

75

Configure an isolated tmux server per OpenRig home with `rig config set terminal.tmux_server isolated` , so multiple homes on the same machine don't share a tmux server.

Isolate each OpenRig home's tmux sessions so that multiple homes on the same machine don't share a server.

```
$ rig config set terminal.tmux_server isolated
```

— Exact config key, value, and runnable command given v0.7.8

THINNER COVERAGE BELOW

05 **Follow a mission from the terminal** NEW

55

Follow a mission from the terminal with `rig terminal open mission:<id> --follow --provider herdr` .

— Exact command and flags given, no example run v0.7.8

06 **Claude seat MCP opt-out** NEW

50

Set `claude_mcp: none` for a Claude seat member to opt that seat out of MCP.

— Exact config key named but no file path or example v0.7.8

07 **Seat activity grant for out-of-setup runners** NEW

45

A seat activity grant lets a runner outside the host's hook setup report activity and launch for one seat without the install-wide hook token.

— Mechanism described in prose, no command or config key v0.7.8

08 **Host management commands added** NEW 40

OpenRig now adds `rig host edit` and `rig host remove` commands for host management.

— Bare command names, no usage detail or example v0.7.8

09 **DeepSeek Harness community runner example** NEW 35

An example DeepSeek Harness (`dsh`) runner with its preset is available under `docs/reference/community-harness-examples/dsh/` as a build reference.

— Path given but no usage example or command v0.7.8

└──▷ **BREAKING ON UPGRADE**

! Restart the daemon after upgrading (`rig daemon stop && rig daemon start`); four database migrations (142–145) run on first start and cannot be skipped.

! The bundled `openrig-core` plugin advances from 0.1.23 to 0.1.24; one line of its seat-continuity skill changed, and managed refresh now stores replaced files under `$OPENRIG_HOME/backups/managed-refresh/<id>/` — reapply any local edits you still need from that copy.

! Runner seats following section 7 of `docs/reference/harness-runner-contract.md` now receive answers or permits instead of automatic refusals; use `--deny` to keep a seat refusing.

WAS THIS USEFUL? 👍 👎

mksglu Context Mode

Feedback ⓘ v2.1.0 NOTES ↗ RANK

Free, local plugin that keeps tool output out of your coding agent's context window. Sandbox and SQLite FTS5 search, 99% fewer bytes in context.

> Context Mode 2.0 rebuilds the tool around nine MCP tools — sandboxed command execution, batch search, URL indexing, and usage stats — that together keep up to 99.4% of raw tool output out of the agent's context window, paired with a large breaking overhaul of its runtime, storage layout, and supported-agent list; v2.1.0 follows with dry-run release plan previews.

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

01 **In-place migration via /ctx-upgrade** NEW 88

`/ctx-upgrade` migrates a 1.x installation in place, replacing old entries while keeping existing hooks, hook timeouts, `permissions.deny`, and MCP entries; running `/ctx-doctor` afterward verifies the result.

Migrate a 1.x installation to 2.0 without losing custom hooks or permissions, then verify the result — run inside the agent chat after restarting it.

```
$ /ctx-upgrade
/ctx-doctor
```

— Runnable commands and exact fields preserved during migration v2.0.0

02 **Sandboxed command execution via ctx_execute** NEW 80

`ctx_execute` runs shell, JavaScript, TypeScript, and Python commands in a local sandbox and returns only printed output to the agent context — a 1,039,898 B JSON payload came back as 254 B. It replaces the removed `ctx_execute_file`, which is now reached via `ctx_execute` with a `file` field.

— Concrete byte-reduction example but no invocation syntax shown v2.0.0

03 **Batch command execution and search via ctx_batch** NEW 73

`ctx_batch` runs multiple commands in one call, indexes their output, and lets you search the results, labeling each result under its own `ctx_batch:<turn>:<n>` source. It replaces the removed `ctx_batch_execute`.

— Names the tool and source format, no runnable example given v2.0.0

04 **Node 24 runtime requirement** BREAKING 73

Context Mode 2.0 requires Node 24.15.0 or later (up from Node 22.5 in 1.x) and uses `node:sqlite` directly; `better-sqlite3` and Bun support are removed.

— Clear version requirement and removed dependencies v2.0.0

05 **Remote URL indexing via ctx_index** NEW 71

`ctx_index` indexes remote content by URL using the `url` field, converting fetched HTML into readable markdown before chunking. It replaces the removed `ctx_fetch_and_index`.

— Mechanism and field named, no example call shown v2.0.0

06 **Offline benchmark suite** NEW 70

Run the benchmark suite offline with `cd bench && node run.mjs` to reproduce the published output-reduction tables on your own machine.

Run the full benchmark suite offline to verify output-reduction numbers on your own machine before deploying to a team.

```
$ cd bench && node run.mjs
```

— Exact runnable command given v2.0.0

07 **Hooks for 10 coding agents** NEW 70

Context Mode ships hooks for 10 coding agents, installed via `context-mode setup --agents <agent>`, for example `npm i -g context-mode && context-mode setup --agents copilot-cli`.

Set up Context Mode hooks for Copilot CLI after installing the npm package — required once before the agent can use any ctx_ tools.

```
$ npm i -g context-mode && context-mode setup --agents copilot-cli
```

— Exact install and setup command given, agent count only vague v2.0.0

08 **Automatic chunking of large tool output** IMPROVED 68

On hosts that run Context Mode first, any output over 4,096 B is stored whole and the agent receives only the best matching chunks — a 666,796 B log put just 3,380 B into context.

— Concrete numbers and mechanism, but it's automatic with no user action v2.0.0

09 **Dropped support for eight coding agents** BREAKING 68

Support for Cursor, JetBrains Copilot, Kimi, Codex Desktop, OpenClaw, OMP (Oh My Pi), Antigravity (agy), and Zed is removed in 2.0; stay on 1.0.169 to keep using those agents.

— Every dropped agent named with a clear fallback version `v2.0.0`

10 **Expanded deny policy for file-dump commands** IMPROVED

65

The shell-dump deny policy that already blocked `cat` now also denies unbounded file-dump commands including PowerShell `Get-Content`, `gc`, `type`, and `[IO.File]::ReadAllText`.

— Names every blocked command but is a passive safety policy `v2.0.0`

11 **Dry-run preview for release plans** NEW

60

Context Mode adds release plan visibility showing which agent and channel receives each release, and a `dry_run` input lets you preview a release's distribution plan before it ships.

— Names the dry_run input but no concrete output shown `v2.1.0`

12 **Per-project context.db storage** BREAKING

60

Each project now uses one `context.db` under `~/.context-mode`; 2.0 starts a new index and does not copy over 1.x data.

— Named path but no migration path for existing data `v2.0.0`

THINNER COVERAGE BELOW

13 **Removed legacy CLI and daemon interfaces** DEPRECATED

58

`ctx_insight` and the `context-mode insight` CLI command are removed, along with the statusline, HTTP daemon mode, and the CLI `index` and `search` commands.

— Every removed surface named, no replacement guidance given `v2.0.0`

14 **Routing moved out of CLAUDE.md/AGENTS.md templates** BREAKING

50

Setup no longer writes `CLAUDE.md` or `AGENTS.md` routing templates into the project; routing now lives in the tool descriptions instead.

— Names the files affected but little mechanism on new routing `v2.0.0`

15 **Context savings summary via ctx_stats** NEW

40

`ctx_stats` reports a per-session and all-time summary of how many output bytes were kept out of the context window.

— Thin one-line description of the tool's purpose v2.0.0

└──▷ BREAKING ON UPGRADE

! Node 24.15.0 or later is required; 1.x required Node 22.5. `node:sqlite` is now used directly and `better-sqlite3` and Bun are removed.

! Each project now uses one `context.db` under `~/.context-mode`; 2.0 starts a new index and does not copy 1.x data.

! `ctx_execute_file` is replaced by `ctx_execute` with a `file` field.

! `ctx_batch_execute` is replaced by `ctx_batch`.

! `ctx_fetch_and_index` is replaced by `ctx_index` with a `url` field.

! `ctx_insight` and the `context-mode insight` CLI command are removed.

! The statusline, HTTP daemon mode, and the CLI `index` and `search` commands are removed.

! Setup no longer writes `CLAUDE.md` or `AGENTS.md` routing templates into the project; routing now lives in the tool descriptions.

! Support for Cursor, JetBrains Copilot, Kimi, Codex Desktop, OpenClaw, OMP (Oh My Pi), Antigravity (agy), and Zed is removed in 2.0; stay on 1.0.169 to keep those agents.

WAS THIS USEFUL? 👍 👎

claude_code_vs

Feedback ⓘ v1.22.0 NOTES ↗ RANK

Bring Claude Code to Visual Studio 2026: A native diff with accept/reject, a live debugger Claude can drive autonomously, Roslyn code navigation, and Test Explorer integration. The IDE half of Claude Code's integration protocol.

> claude_code_vs adds a true MSBuild/NuGet resolution tool for the agent, lets you edit Claude's proposed diffs before accepting them, surfaces a live activity card during sessions, and adds a Roslyn-powered Generate Tests command to the editor context menu.

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

01 MSBuild/NuGet project graph tool for the agent NEW

95

A new `vs_solution_model` tool (`vs-semantic`) returns what MSBuild and NuGet actually resolved — real target frameworks, packages with resolved versions split into direct and transitive, project references, active configuration/platform, startup projects, and which `Directory.Build.*` files each project inherits — instead of reading `.csproj` text. Passing a `project` argument returns the full transitive closure with `requestedBy` chains, the analyzer set grouped by origin, and evaluated MSBuild properties that most often diverge from the project file; a `properties` argument evaluates any other MSBuild property by name.

Inspect the full resolved dependency graph of a project — including centrally managed package versions and transitive closure — when diagnosing a build that works differently than the .csproj suggests.

📍 In the Claude Code panel, ask: 'What packages does Pricing.App actually compile against?' — the extension calls `vs_solution_model` with `project: Pricing.App` and returns resolved versions, `requestedBy` chains, and the evaluated MSBuild properties from `Directory.Build.props` .

— Names the tool and both arguments with exact behavior v1.22.0

02 Generate Tests context menu command NEW

90

A Generate Tests entry is added as the seventh item in the Claude Code editor context menu flyout — it resolves the function at the caret through Roslyn, detects the test framework (xUnit, NUnit, MSTest, or TUnit) from what each project compiles against, checks whether the test project already references the code under test and which existing tests call the member, then requests `vs_build` + `vs_run_test` verification of the new tests.

Generate tests for a specific function and have Claude verify they build and pass, without duplicating tests that already cover the member.

📍 Place the caret on a method in a non-test file, right-click, and choose Claude Code › Generate Tests. The extension resolves the function via Roslyn, detects the test framework (xUnit, NUnit, MSTest, or TUnit), and asks Claude to write tests followed by `vs_build` + `vs_run_test` verification.

— Exact menu path, frameworks detected, and verification tools named v1.22.0

03 **Editable diff pane before accepting Claude's edits** IMPROVED

63

The review diff's right pane is now a live editor, so you can rename a symbol or remove a line before clicking Accept; the bridge rebuilds the CLI's Edit/Write input from the file as it is plus the pane as you left it, and Claude is told the file differs from its proposal so it re-reads before touching it again.

Correct a rename or remove an unintended line from Claude's proposed edit before it touches the file — keeping the CLI as the sole writer.

📍 When a diff review appears, edit the right pane directly (rename the symbol, delete the unwanted line), then click Accept. The bridge rebuilds the CLI's Edit / Write input from the file as it is and the pane as you left it; Claude is notified the file differs from its proposal and re-reads before its next edit.

— Explains mechanism but no code-level surface named v1.22.0

THINNER COVERAGE BELOW

04 **Live activity card during sessions** NEW

50

A live activity card in the panel shows elapsed time, the current prompt, and the tool call in flight (e.g. `⚙ Edit Pricing.cs` or `⚙ vs_build BuildBreak.App`), plus Claude's task list when the session has one; it settles to a summary like 'Finished · 2m 05s · 14 tool calls' between turns.

— Describes display states but nothing to configure or invoke v1.22.0

WAS THIS USEFUL? 👍 👎

intelligent-terminal

Feedback ⓘ v0.3 NOTES ↗

A fork of Windows Terminal with native agent integration, right in your command line.

Intelligent Terminal v0.3 adds a collapsible sidebar, headless tab mode, auto-detected agent CLI profiles, keyboard navigation shortcuts, tab pinning, and Markdown replies in the Agent Pane.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.3 https://github.com/microsoft/intelligent-terminal.git
# already have the repo? check out this version:
$ git checkout v0.3
```

Navigate the sidebar with `Ctrl+Shift+S` to enter or toggle it, `Ctrl+Alt+Up/Down` to cycle open tabs, `Ctrl+Shift+G` for Agents-only view, and `Ctrl+Shift+/` for recent sessions.

Search across open tabs, panes, and recent agent sessions from the sidebar with `Ctrl+F`.

Switch between horizontal tabs and a collapsible sidebar without restarting — right-click a tab and select `Switch to horizontal tabs`, or change the layout in Settings.

Keep a tab's shells and agent panes alive after closing its window by choosing `Turn on headless mode` from its context menu; restore or close retained tabs from the system tray, including `Close all keep-running tabs`.

NEW INTEGRATION Open installed agent CLIs (Copilot, Claude, Codex, Gemini, OpenCode, Antigravity) directly from the new-tab menu as auto-detected profiles, independently of the Agent Pane ACP connection.

+7 more

WAS THIS USEFUL? 👍 👎

agentty

Feedback ⓘ v0.9.46 NOTES ↗ RANK

AI pair programming in your terminal — one static binary, sub-ms startup, any model

> agentty 0.9.46 introduces an Allowed Hosts setting that restricts sandboxed commands to approved network destinations, routing all macOS sandbox traffic through a filtering proxy, and further locks down the macOS sandbox against clipboard, keychain, and file-path snooping.

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

01 Allowed Hosts network restriction for sandboxed commands

82

BREAKING

A new Settings → Sandbox → Network → Allowed hosts setting restricts sandboxed commands to a user-defined list of sites, accepting exact hostnames like `github.com` or wildcard patterns like `*.github.com`; pasted URLs are trimmed to the host and invalid entries (typos, IP addresses) are rejected with a reason. On macOS, sandboxed commands no longer reach the network directly — all traffic now routes through agentty's filtering proxy, so any command relying on direct local network or `localhost` access will be refused.

Lock a sandboxed session so the agent can only reach your package registries and version-control host, not arbitrary internet endpoints.

📍 Settings → Sandbox → Network → Allowed hosts Add entries: `registry.npmjs.org`, *.github.com, pypi.org

— Names exact setting path, host patterns, and breaking network behavior. v0.9.46

THINNER COVERAGE BELOW

02 macOS sandbox blocks clipboard, keychain, and path probing

IMPROVED

38

The macOS sandbox now blocks sandboxed commands from reading the clipboard, listing keychains, or probing file paths outside granted locations.

— Lists what's blocked but no mechanism or config surface. v0.9.46

└──▷ **BREAKING ON UPGRADE**

! On macOS, sandboxed commands no longer reach the network directly — all traffic now goes through agentty's filtering proxy, so any command that relied on direct local network or `localhost` access will be refused.

WAS THIS USEFUL? 👍 👎

Google gemini-cli

Feedback ⓘ v0.67.0 NOTES ↗ RANK

An open-source command-line AI agent for coding, task automation, and tool use with Gemini models.

> gemini-cli adds a command-line way to discover available models and extends IDE companion support into sandboxed container environments.

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

01 Model discovery via CLI NEW

75

Discover available models at the command line with the new `models` command or the `--list-models` flag, letting users enumerate models available to their current auth context interactively or in scripts/CI pipelines.

List all models available to your current auth context before switching to a different one.

```
$ gemini models
```

Quickly enumerate available models in a non-interactive script or CI pipeline.

```
$ gemini --list-models
```

— Names exact command and flag with runnable examples v0.67.0

THINNER COVERAGE BELOW

02 IDE companion in container sandboxes IMPROVED

45

Bridges the IDE companion into container sandboxes over exec stdio, enabling IDE integration inside sandboxed environments.

— Explains mechanism but no concrete usage steps given v0.67.0

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ desktop-v2.0.4 NOTES ↗ RANK

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

> Superset's desktop app adds full browser-extension support (verified loading, `declarativeNetRequest`, `storage.sync`) and a redesigned shell behind the `paradev-shell` flag, alongside a Linear integration, archived automations, new plugins (Inngest, Supabase, Firecrawl, Wispr Flow), and a breaking move of the app home directory from `~/.superset` to `~/.paradev`.

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

01 Browser extension support in browser pane NEW

95

Adds the `superset browser extensions` CLI command for managing browser pane extensions, verifies extension files against Web Store hashes before loading, enforces `declarativeNetRequest` so content-blocker extensions can block network requests, and supports `storage.sync`, `onInstalled`, and persistent context menus. Also adds support for Chrome extensions and passkeys in the browser pane.

List and manage Chrome extensions installed in the desktop browser pane without opening the UI.

```
$ superset browser extensions
```

— Names CLI command, APIs, and verification mechanism. desktop-v2.0.4

02 Redesigned desktop shell with paradev-shell flag NEW

75

Redesign the desktop shell behind the `paradev-shell` flag, adding a rail, title-bar tabs, pane cards, and a composer home. Enable it in Settings to switch to the new shell layout.

Enable the redesigned desktop shell with the new rail, title-bar tabs, and pane cards.

> 📍 In the desktop app, open Settings and enable the `paradev-shell` flag to switch to the new shell layout.

— Names flag and exact UI elements with enablement steps. desktop-v2.0.4

03 App home directory moved to ~/.paradev BREAKING

70

The app home directory moves from `~/.superset` to `~/.paradev`; repo config in `.paradev/` is now canonical, with `.superset/` still read as a fallback for backward compatibility.

— Exact paths given; no migration command. desktop-v2.0.4

THINNER COVERAGE BELOW

04 **Quick-pick settings in new-workspace composer** NEW

50

Set project, host, environment, and size in the new-workspace composer using `#` as a quick-pick prefix.

— Names exact prefix but no further mechanism. desktop-v2.0.4

05 **Linear project issues in channels** NEW

50

Show a Linear project's issues inside a channel and let you pick or create a channel's project from its Tasks tab.

— Names Tasks tab but no further integration detail. desktop-v2.0.4

06 **Archive automations instead of delete** IMPROVED

40

Archive automations instead of deleting them, available from both the desktop UI and CLI.

— States behavior change but no exact command names. desktop-v2.0.4

07 **Inngest, Supabase, Firecrawl, and Wispr Flow plugins** NEW

35

Add Inngest, Supabase, Firecrawl, and Wispr Flow plugins to the plugin catalogue.

— Names plugins but no detail on capabilities. desktop-v2.0.4

08 **Pinned pages in sidebar** NEW

30

Show pinned pages in the sidebar's Pinned section.

— Names UI location only. desktop-v2.0.4

09 **Can-edit role for page sharing** NEW

30

Share pages with an Can edit role in page sharing.

— Names role but no further detail. desktop-v2.0.4

10 **Subagents and background tasks shown in chat** NEW

20

Show running subagents and background tasks above the chat composer.

— Minimal description, no mechanism. desktop-v2.0.4

└──▷ **BREAKING ON UPGRADE**

! The app home directory moves from `~/.superset` to `~/.paradev`; repo config should be placed in `.paradev/` going forward (`.superset/` is still read but is the legacy path).

WAS THIS USEFUL?  

zentty

Feedback ⓘ v0.6.0 NOTES ↗ RANK

A native macOS terminal for agent-driven development, built on Ghostty.

> zentty v0.6.0 adds searchable, resumable Agent History for past AI coding sessions and detachable floating panes, alongside new clipboard and link-safety confirmations.

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

01 **Agent History search and resume** NEW 85

Search, preview, and resume past Claude Code, Codex, Grok, and OpenCode conversations via Agent History (⌘⇧H) or the `zentty history` command, letting agents running in a pane retrieve prior sessions directly. History indexing is opt-in, with history and Claude retention controls available in Settings.

Let an agent running in a pane pull up a previous session to continue work without re-explaining context.

```
$ zentty history
```

Open Agent History from the keyboard to search and resume a past Claude Code or Codex conversation.

📍 Press ⌘⇧H to open Agent History, search for the session by keyword or date, then select it to resume.

— Names command, shortcut, supported agents and settings controls v0.6.0

02 **Floating panes with docking and previews** NEW 80

Create floating panes using `zentty float` commands, with support for dragging, resizing, docking, pill previews, and show/hide controls for detaching a pane into its own window.

Detach a pane into a floating window that shrinks to picture-in-picture while you work in another space.

```
$ zentty float
```

— Runnable command plus named interaction features v0.6.0

THINNER COVERAGE BELOW

03 **Clipboard read confirmation** NEW 30

Terminal programs must now obtain user confirmation before reading clipboard contents, a new security gate on clipboard access.

— Describes behavior but no config or command surface v0.6.0

04 **Safety checks for untrusted links** NEW 25

Adds safety checks for untrusted links opened with Command-click, guarding against malicious links encountered in terminal output.

— Brief description with no mechanism or config detail v0.6.0

WAS THIS USEFUL? 👍 👎

zot

Feedback ⓘ v0.4.63 NOTES ↗ RANK

zot manages coding agent workflows from the command line.

> zot adds CloudGPU as a new built-in provider, expanding its roster of supported AI providers to over 30.

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

01 **CloudGPU provider support** NEW 45

zot adds CloudGPU as a built-in provider, selectable alongside existing providers such as Anthropic, OpenAI, Gemini, and Ollama, bringing the total built-in provider count to 30+.

— Names the provider but no config or selection mechanism given. v0.4.63

WAS THIS USEFUL? 👍 👎

forgedock

Feedback ⓘ v1.48.0 NOTES ↗ RANK

ForgeDock — autonomous AI dev pipeline for Claude Code. GitHub as agent memory: issue in, PR out.

> ForgeDock's issue-watch now trusts issues authored by the repo's own ForgeDock App identity, fixing a case where watches were skipped.

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

01 **Trust for ForgeDock App-authored issues in issue-watch**

IMPROVED 45

Issue-watch now opts in to trusting issues authored by the repo's own ForgeDock App, so watches are no longer skipped when `gh` runs as the App identity.

— Names the mechanism and condition but no config surface or command v1.48.0

WAS THIS USEFUL? 👍 👎

devoxxgenieideaplugin

Feedback ⓘ v1.17.0 NOTES ↗

RANK

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 v1.17.0 adds configurable RAG embeddings across multiple LLM providers with a dedicated retrieval benchmark command, plus in-IDE testing tools for RAG queries and web search credentials.

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

01 **RAG retrieval benchmark command** NEW 95

Run `./gradlew ragBenchmark` (also available as `RagCli benchmark`) to measure RAG retrieval quality: it indexes the repository in memory, runs 28 built-in questions with known answer files, and reports recall@1/5/10, MRR, and per-question pass/fail, with optional JSON output — without touching ChromaDB.

Measure how well your current RAG embedding setup retrieves answers across the built-in question set before switching providers or tuning dimensions.

```
$ ./gradlew ragBenchmark
```

— Exact runnable command with concrete metrics and output options v1.17.0

02 **RAG embeddings via configurable LLM providers** NEW 78

RAG embeddings can now run through enabled LLM providers — Ollama, Custom OpenAI, OpenAI, LM Studio, and llama.cpp — reusing the URLs and keys already set in LLM settings. `nomic-embed-text` gets its prefixes automatically, and users can set query prefix, document prefix, and output dimension for embeddings (prefixes are never stored or shown to the LLM); the index now binds to provider, model, dimension, and document prefix so search keeps working until a full re-index runs.

— Names providers and settings but no exact config keys or UI path v1.17.0

03 **In-settings RAG and web search testing** NEW 65

Run a live test query against the project's RAG index directly from the RAG settings page, and test web search credentials (Tavily or Google) from the web search settings page; both run in the background with results or errors shown inline.

Validate RAG settings and check live retrieval results without leaving the IDE settings panel.

> 📍 In the IDE, go to Settings › DevoxxGenie › RAG Settings and click `Test Query` to run a search against the project index and see ranked matches with scores.

— Clear UI navigation given, but no deeper mechanism described v1.17.0

---

04 **RagCli and WebSearchCli removed from plugin bundle** BREAKING

60

The `RagCli` and `WebSearchCli` developer tools move to test sources and no longer ship inside the plugin; the `ragQuery` and `webSearch` Gradle tasks now use the test runtime classpath.

— Names exact tasks and tools affected but no migration steps v1.17.0

└──▷ **BREAKING ON UPGRADE**

! The `RagCli` and `WebSearchCli` developer tools move to test sources and no longer ship inside the plugin; the `ragQuery` and `webSearch` Gradle tasks now use the test runtime classpath.

WAS THIS USEFUL? 👍 👎

coffee-cli

Feedback ⓘ v3.9.1 NOTES ↗ RANK

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

> coffee-cli's biggest addition is Pi desktop conversation mode with live diffs and session metrics, alongside Pi's expansion into multi-agent orchestration and automation, plus a new sidebar tool filter and worktree/window refinements.

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

01 **Pi desktop conversation mode** NEW 70

Choosing Desktop on the Pi card in the launchpad gives side-by-side code diffs, streaming command output, inline model and thinking-level switching, and a live view of context use, cache hit rate, and session cost. Pi desktop conversations now also have access to the composer's Worktree switch, matching the capability already available in other desktop modes.

– Names UI path and feature set but no exact flags or commands. v3.9.1 v3.8.9

THINNER COVERAGE BELOW

02 **'Tools' filter for sidebar sessions** NEW 55

A new Tools option in the three-line menu above the sidebar list lets users filter sessions by a single tool, with worker sessions still grouped under their main session.

– Clear UI path but thin on mechanism beyond filtering. v3.9.1

03 **Pi multi-agent orchestration and automation controls** NEW 50

Pi now works as a worker session in multi-agent orchestration, enabled in app settings, and supports browser control and computer control, with its usage now counted in usage statistics.

– Describes capability and settings location but not exact steps. v3.8.9

04 **Safer worktree merge handling** IMPROVED 40

Worktree merges now check the main checkout first and skip it when a merge or rebase is already under way, and newly created worktrees are no longer cleaned up by mistake.

– Explains before/after behavior but no user-facing action. v3.9.1

05 **Window size and position persistence** IMPROVED 30

Window size and position are now remembered across restarts, and the default window opens smaller to fit 13-inch laptops.

— Simple UX change with no configurable surface named. v3.9.1

WAS THIS USEFUL? 👍 👎

Zed

Feedback ⓘ v1.27.0 NOTES ↗

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

Zed adds a setting to open README files by default on new windows, and AI commit messages now mirror your repo's recent commit style.

└──▷ GET THIS VERSION ▾

Homebrew

```
$ brew install --cask zed
# already installed? upgrade to this version:
$ brew upgrade --cask zed
```

git

```
$ git clone --branch v1.27.0 https://github.com/zed-industries/zed.git
# already have the repo? check out this version:
$ git checkout v1.27.0
```

Homebrew lists this version · checked 5 Nov Wrong command?

Open README files automatically on new windows by setting `on_new_window` to `"readme"` in your config.

AI commit message generation now uses the three most recent repository commit messages as style examples.

RETIRED SUPPORTED PLATFORM macOS Catalina and Big Sur support

NEW SUPPORTED PLATFORM Go 1.27 generic methods syntax highlighting

└──▷ BREAKING ON UPGRADE

! Support for macOS Catalina and Big Sur has been removed; Zed now requires macOS 12.0 (Monterey) or later.

! Agent checkpoints and the Restore Checkpoint button have been removed; use edit rejection or Git to undo agent changes.

WAS THIS USEFUL? 👍 👎

swttch

Feedback ⓘ v0.35.2 NOTES ↗ RANK

Claude Code GUI plugin for JetBrains IDEs. (ex - Claude Code with GUI)

> swttch added a configurable context badge display option in v0.35.2.

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

01 **Configurable context badge display** IMPROVED 35

The context badge can now be set to show percent, token count, or nothing, giving users control over how context usage is displayed.

— Names the three display options but no exact UI path or setting name v0.35.2

WAS THIS USEFUL? 👍 👎

agents-cli

Feedback ⓘ v1.13.0 NOTES ↗ RANK

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

> agents-cli rounded out its agent lifecycle toolchain this window with Agent Identity for Cloud Run deployments, enforceable eval pass/fail thresholds with a new breaking exit-code behavior, and a cluster of scaffold/deploy additions covering pre-built Agent Runtime images, GKE dry-run validation, extension pinning, and app-name/region tracking in the project manifest.

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

01 Eval pass/fail thresholds and reporting BREAKING 91

`agents-cli eval run` executes `eval generate` then `eval grade` and reports a numeric quality score against a configurable pass threshold (e.g. `custom_response_quality 4.71 >= 4.50`). Minimum mean scores per metric can now be set via a `thresholds:` map in the eval config or per run with `agents-cli eval run --fail-under METRIC=SCORE` (repeatable); missed thresholds exit 1, and both `agents-cli eval run` and `agents-cli eval grade` end with a `RESULT: PASS` or RESULT: FAIL (...) line. A run where any case errored now also exits 1 even with no thresholds set — previously errors were non-fatal; if the errors are rate limits, lower `--qps`.

Fail CI when eval quality drops below defined per-metric score floors, so a regression in accuracy or groundedness blocks the deployment pipeline.

```
$ agents-cli eval run --fail-under trajectory_match=0.8 --fail-under groundedness=0.75
```

Set persistent eval score thresholds in the project eval config so every run enforces them without requiring CLI flags.

```yaml
thresholds:
  trajectory_match: 0.8
  groundedness: 0.75
```

— Names flags, config key, and exact exit-code behavior change. v1.13.0

02 New deploy command capabilities NEW 86

Adds several new `deploy` command capabilities: declaring an `agents-cli-extension.yaml` with a `commands.override.deploy.run` block lets teams run their own script before every `deploy` invocation, enforcing policies without changing each caller's workflow; `agents-cli deploy --agent-identity` deploys a Cloud Run service using Agent Identity, which an existing project can adopt via `agents-cli scaffold enhance --cloud-run-identity-type agent_identity`, or a new project can instead use a service account via `agents-cli scaffold create --cloud-run-identity-type service_account`; `agents-cli deploy --image <artifact-registry-uri>` deploys an Agent Runtime service from a pre-built image instead of building from source, requiring `roles/artifactregistry.reader` on the repository for the Agent Runtime service agent, and `--build-args` cannot be combined with `--image`; and Agent Runtime deploys of A2A agents now register the agent card in Agent Registry, making the agent visible as A2A-capable in Gemini Enterprise.

Enforce a team policy by running a gate script before every deploy across all teams and CI, without changing each caller's workflow.

```yaml
commands:
  override:
    deploy:
      run:
        - "bash"
        - "policy.sh"
```

Deploy an Agent Runtime service from a pre-built Artifact Registry image instead of a source build, useful when the image is built in a separate CI pipeline.

```
$ agents-cli deploy --image us-docker.pkg.dev/my-project/my-repo/my-agent:v1.2.3
```

— Named config file and override block with example. Three concrete flag paths cover new, migrated, and service-account cases. Exact flag, IAM role requirement, and exclusion rule plus runnable example. Describes outcome but no command or setup step given.

v1.13.0

03 **Agent project scaffolding with framework templates** NEW

75

`agents-cli create <name> --agent adk_python --prototype` generates an app directory and a working tool stub ready for evaluation. ADK (Python or Go), LangGraph, and Claude Agent SDK are supported as first-party framework templates, with a bring-your-own-template option.

Bootstrap a new agent project, evaluate its quality against a numeric threshold, and deploy to Cloud Run in one workflow.

```
$ agents-cli create support-agent --agent adk_python --prototype
agents-cli eval run
agents-cli scaffold enhance --deployment-target cloud_run
agents-cli deploy --project my-project
```

— Exact scaffold command and supported frameworks both named. product docs

---

04 **Extension version pinning and trusted installs** IMPROVED

73

`agents-cli extension add <name>` now pins a first-party extension to the latest compatible release instead of `main`; pass `--ref main` to track `main`, and move an existing pin with `agents-cli extension update --to <ref|:latest>`. `agents-cli extension add` also installs extensions from pre-approved repositories without a trust prompt, whether given as a shorthand, an `org/repo` reference, or a URL.

— Named commands and flags for pinning and trust bypass. v1.13.0

---

05 **Scaffold enhance deployment target and region flags** IMPROVED

62

`agents-cli scaffold enhance --deployment-target cloud_run` adds a deployment target to an existing project before deploying. `agents-cli scaffold enhance --region <region>` now writes the region to `agents-cli-manifest.yaml` and to rendered files such as Terraform variables.

Bootstrap a new agent project, evaluate its quality against a numeric threshold, and deploy to Cloud Run in one workflow.

```
$ agents-cli create support-agent --agent adk_python --prototype
agents-cli eval run
agents-cli scaffold enhance --deployment-target cloud_run
agents-cli deploy --project my-project
```

— Two named flags with their effect on manifest and config files. v1.13.0

THINNER COVERAGE BELOW

---

06 **App name recorded in project manifest** NEW

57

New projects record `create_params.app_name` in `agents-cli-manifest.yaml` for the agent's served name (`/apps/{app_name}/...`), and `eval generate --app-name` and `eval run --app-name` now default to the project's app name.

— Names config key and flags but no usage example shown. v1.13.0

07 **Live agent test requests with trace capture** NEW 55

`agents-cli run --url <endpoint> --mode a2a` sends a test request to a live agent and captures its trace, letting you confirm tool calls are executing against a deployed endpoint.

Send a live A2A test request to a deployed agent and inspect its Cloud Trace output to confirm tool calls are executing.

```
$ agents-cli run --url https://support-agent-482910375521.us-east1.run.app --mode a2a "Where is order 1042?"
```

— Exact command given but mechanism of trace capture unexplained. product docs

08 **Combined gcloud and ADC sign-in** IMPROVED 50

`agents-cli login` now signs in both the gcloud CLI and Application Default Credentials in one step (equivalent to `gcloud auth login --update-adc`), so gcloud commands work immediately after login.

— States mechanism and equivalent command but no new flag. v1.13.0

09 **Dry-run validation for GKE deploys** IMPROVED 30

`agents-cli deploy --dry-run` now also validates GKE deployments against Google Cloud.

— Single-line addition with no further mechanism or example. v1.13.0

└──▷ **BREAKING ON UPGRADE**

! An `agents-cli eval run` or `agents-cli eval grade` run where any case errored now exits 1, even when no `thresholds:` are set. Previously, errors were non-fatal. If the errors are rate limits, lower `--qps`.

WAS THIS USEFUL? 👍 👎

pstack-claude

Feedback ⓘ v0.9.129 NOTES ↗ RANK

pstack-claude is a port of Poteto's pstack agent framework to Claude Code, Codex, Copilot, Pi, and other AI coding assistants.

> pstack-claude added full support for Google's Antigravity IDE, bringing its skill/agent framework and tool-routing hooks to a new assistant platform.

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

01 **Antigravity platform support for pstack** NEW 94

pstack can now be installed on Antigravity via `agy plugin install ./plugins/pstack`, registering 67 skills, 4 agents, and a `SessionStart` routing hook. The hook runs `./hooks/session-start.sh agy` and emits the poteto-mode mandate as an `injectSteps` system message; because Antigravity's native skill loader retains only the first 24,000 bytes of a skill body, the hook instructs the agent to read poteto-mode's `SKILL.md` in full via `read_file`. The poteto-mode mandate can be disabled by setting `session hook: off` in `~/.gemini/config/pstack-models.md`, and a new `agy-tools.md` reference maps Claude tools to Antigravity equivalents, documenting `invoke_subagent` with `Model` values `inherit`, `flash_lite`, `flash`, and `pro`.

— Exact install command, config path, hook internals and tool mapping all given. v0.9.129

WAS THIS USEFUL? 👍 👎

palot

Feedback ⓘ v0.23.0 NOTES ↗ RANK

Just another desktop client for OpenCode 2. Manage projects, sessions, parallel agents, requests, and changes on Linux and macOS.

> palot v0.23.0 ships its first downloadable desktop builds for Linux and macOS, along with package verification support and a note that auto-updates are not yet wired up.

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

01 **Package integrity verification** NEW 73

Desktop packages can be verified with `SHA256SUMS` or by using `gh attestation verify` against GitHub build attestations.

— Names exact verification command and checksum file v0.23.0

02 **Linux and macOS desktop packages** NEW 63

palot now adds Linux x64 desktop packages in `AppImage`, Debian/Ubuntu `DEB`, Fedora `RPM`, and portable `tar.gz` formats, plus macOS Apple Silicon packages in `DMG` and `ZIP` formats (ad-hoc signed; not Developer ID signed or notarized).

— Lists concrete formats but no install steps beyond download v0.23.0

THINNER COVERAGE BELOW

03 **No auto-update feeds published** BREAKING 50

No auto-update feeds are published in this release — updates must be applied manually, and no legacy Palot 0.11 auto-update feeds are served.

— States the change but no migration guidance given v0.23.0

└──▷ **BREAKING ON UPGRADE**

! No auto-update feeds are published — updates are manual and no legacy Palot 0.11 auto-update feeds are served.

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ

announced · Nov 4, 2026 BLOG ↗ RANK

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

> Graphify added MCP server discovery endpoints and standards-based OAuth authentication for its remote MCP server, making it easier for AI clients and registries to find and securely connect to the service.

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

01 MCP server discovery via server card and AI Catalog NEW

81

Graphify now publishes a public MCP server card at `https://api.graphify.com/mcp/server-card` exposing the server's name, icon, and remote URL so AI clients can discover it without manual config, plus an AI Catalog index at `https://api.graphify.com/.well-known/ai-catalog.json` that lets any client or registry knowing only the domain locate that server card automatically. Clients such as Claude Code can connect to the remote MCP server with `claude mcp add --transport http --scope user graphify https://api.graphify.com/mcp`.

Inspect Graphify's MCP server card to see its registry name, remote URL, and repository before connecting any client.

```
$ curl -s https://api.graphify.com/mcp/server-card | jq '{name, title, remotes, websiteUrl, repository}'
```

Check the AI Catalog to confirm the domain's published MCP artifacts and retrieve the server card URL programmatically.

```
$ curl -s https://api.graphify.com/.well-known/ai-catalog.json | jq
```

Add Graphify as a remote MCP server in Claude Code using the streamable-HTTP transport so discovery-aware clients fill in the rest automatically.

```
$ claude mcp add --transport http --scope user graphify https://api.graphify.com/mcp
```

— Two endpoints and a connect command given verbatim with runnable examples.

launch-20261104-393cc757

02 **OAuth PKCE authentication on MCP endpoint** NEW 70

Graphify's MCP endpoint now returns a `401` with a `www-authenticate` header pointing to `https://api.graphify.com/.well-known/oauth-protected-resource/mcp`, enabling standards-based OAuth PKCE sign-in without needing a client secret.

— Named status code, header, and discovery URL but no walkthrough command.

launch-20261104-393cc757

WAS THIS USEFUL? 👍 👎

Amazon Kiro Feedback ⓘ

General: Share credits across your team with credit pooling (2026-11-04) BLOG ↗

RANK

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

> Kiro's two releases this window both center on credit pooling for Enterprise customers, letting organizations share one monthly credit balance across seats instead of enforcing individual per-seat limits, backed by granular usage controls, group management, dashboards, and add-on credit purchases.

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

01 Credit pooling for Enterprise accounts BREAKING 88

Enterprise accounts can enable credit sharing via the Credit sharing setting in the Kiro console, where each seat contributes 500 monthly credits to one shared pool at $20 per seat per month, letting power users draw on capacity that lighter users leave unused. Existing individually-subscribed users can be migrated to the shared pool with the Switch to credit sharing wizard, which reviews a prorated first-month cost and cannot be undone once confirmed. Credit pooling is now the only option for new Kiro profiles — individual per-seat subscriptions are no longer available for new accounts.

Turn on credit pooling for your enterprise org so no single user's unused credits go to waste at month end.

> 📍 In the Kiro console, go to your enterprise admin view, find Credit sharing, and enable it. Each seat added will contribute 500 credits per month to the shared pool.

Move existing individually-subscribed users onto the shared pool without losing their current seat allocation.

> 📍 In the Kiro console, open the Switch to credit sharing wizard, select the users or groups to migrate, review the prorated first-month cost, and confirm. Note: this action cannot be undone.

— Names exact UI setting, wizard, pricing and seat allowance

General: Share credits across your team… launch-20261104-e9974d01

02 Per-user, per-group and per-profile usage limits with threshold alerts

NEW 80

Administrators can set usage limits at the profile, group, and individual level; when a user belongs to multiple groups the highest group limit applies, and further usage is blocked once a limit is hit. Kiro sends configurable threshold alerts to administrators when credit utilization reaches defined levels, defaulting to 80% and 100% firing for per-user limits, pool credit limits, and seat utilization.

Before a heavy sprint, set a per-group usage limit so a single team cannot exhaust the shared pool overnight without an administrator noticing.

📍 In the Kiro admin console, go to Groups, select the target group, open Usage limits, set a group-level credit limit, then confirm the threshold alert is configured to fire at 80% and 100% utilization.

— Specific limit levels and alert thresholds, with worked example

launch-20261104-e9974d01 | General: Share credits across your team…

---

03 **Group management with AWS IAM Identity Center sync** NEW

70

Groups for the shared credit pool can be created manually or populated via AWS IAM Identity Center (IDC) for automatic seat reconciliation, with mid-cycle seat additions credited to the pool immediately.

When onboarding a new Enterprise team mid-cycle, enroll members via AWS IAM Identity Center so their seats and credits are added to the pool immediately without manual rebalancing.

📍 In the Kiro admin console, go to Groups, select Create group, choose Populate via AWS IAM Identity Center (IDC), select the IAM Identity Center group, and save. New seats are pro-rated and credits are added to the shared pool instantly.

— Names exact IDC integration with step-by-step admin console path

launch-20261104-e9974d01

THINNER COVERAGE BELOW

04 **Pool-level usage dashboards** NEW

45

Pool-level dashboards give administrators a real-time view of total credits available, consumed, and remaining across the organization, with seat utilization tracking to identify underused seats.

— Describes dashboard scope but no exact navigation given launch-20261104-e9974d01

---

05 **Add-on credit purchases for the shared pool** NEW

45

Organizations can buy additional credits to top up the shared pool; add-on credits roll over for up to 12 months from purchase, allowing capacity to be bought in a single procurement cycle.

— Gives rollover window but no purchase flow detail `launch-20261104-e9974d01`

`General: Share credits across your team…`

└──▷ **BREAKING ON UPGRADE**

! Credit pooling is the only option for new Kiro profiles — individual per-seat subscriptions are no longer available for new accounts.

! Switching existing users to credit sharing via the `Switch to credit sharing` wizard cannot be undone.

WAS THIS USEFUL?  

◆ Model Training & Fine-Tuning

fizgig

Feedback ⓘ v9.0.4 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's latest releases add full training and fine-tune support for BRIA's Fibo 2.0 text-to-image model, three new optimizers, masked training for MiniMax H3 stills, transparent PNG handling across training surfaces, and several smaller Studio and UI refinements.

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

01 **Fibo 2.0 text-to-image training support** NEW 90

Fizgig adds support for BRIA's Fibo 2.0 8.2B text-to-image model for LoRAs, edit LoRAs, slider LoRAs and full fine-tunes, trainable from a 12 GB card. Setup is via the Preferences tab's Download models for me button, which fetches ~28 GB from Comfy-Org with no Hugging Face token required. Training previews render in 4 steps using Comfy-Org's Turbo LoRA, and all saved files — including fine-tunes — load directly in ComfyUI. Fibo 2.0 supports masked training, Studio with Turbo Preview, LoRA Royale with seed and prompt travel, the Profiler and Extract. Its LoRA preset uses Automagic from `1e-6` for 40 epochs; base precision Auto trains INT8 on 16 GB cards and NF4 on 12 GB cards, with an int8 DiT build offered alongside the bf16 one.

— Names mechanism, presets, VRAM tiers and exact setup path v9.0.4

02 **Prodigy, Muon and Automagic v4 optimizers** NEW 70

Fizgig adds three optimizers — Prodigy, Muon, and Automagic v4 — selectable under Optimizer on the Training tab; Prodigy and Automagic v4 set their own learning rate and disable Adaptive LR automatically.

— Names exact optimizers and UI location, limited mechanism depth v9.0.3

03 **Masked training for MiniMax H3 stills** NEW 70

Fizgig adds masked training for MiniMax H3 stills, introducing a Masked training card with prior preservation outside the mask, at roughly 6% extra training time for masks and 39% for prior preservation with no extra GPU memory.

— Gives concrete overhead numbers and feature location v9.0.3

04 **Transparent PNG support across training surfaces** NEW 70

Transparent PNGs now train with their alpha channel preserved on Qwen Image 2.1, with checkerboard previews; on all other models transparent PNGs are composited on white to match what the captioner saw, applied across face finding, Make masks, the Look filter, Studio, LoRA Royale, and the Profiler.

— Explains model-specific behaviour and lists affected surfaces v9.0.3

THINNER COVERAGE BELOW

05 **Right-click text editing in app window** NEW 40

Right-click now works in the app window, adding Cut, Copy, Paste, and Select all in text boxes, and Copy on selected text.

— Lists exact actions added but minor scope v9.0.3

06 **Expanded loss graph metrics** IMPROVED 35

The loss graph now shows the epoch average, a running average, and the live learning rate.

— Names the metrics but no usage path given v9.0.3

07 **Direct value entry in Studio blocks** IMPROVED 30

Studio now lets you type a block's value directly into its box.

— Bare one-line UI tweak with no further detail v9.0.3

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Databricks Omnigent

Feedback ⓘ announced · Nov 5, 2026

BLOG ↗ RANK

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

> Omnigent's headline addition is full interactive GitHub pull-request management inside coding-agent sessions — edit, review, and merge without leaving the session — alongside Codex thread import with compaction-aware history, SSH remote sessions, and inline GitHub review conversations in the Changes view; it also raises its OpenCode and macOS minimum version requirements.

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

01 Interactive GitHub pull-request management in sessions NEW

90

In the Pull Requests tab of a coding-agent session you can edit a PR's title and description, add or remove labels, and request or cancel reviewer requests, pasting screenshots or videos directly into the description. Comments support a Markdown preview with drafts preserved across tab/PR switches, and hovering the check count expands individual CI jobs with links to their runs. From the Summary tab you can merge using any method the repo allows (merge, squash, or rebase) or turn off auto-merge; Omnigent refuses the merge if the agent pushed new commits after review started. Any GitHub PR can be linked to a session via `+` (Link a PR) by pasting its URL, and PRs the agent opened appear automatically. Actions run on the session's host as the PR's selected GitHub account (default: the host's active `gh` account) and require edit access to the session; GitLab merge requests and Azure DevOps pull requests now also appear in the tab, but read-only.

Review a GitHub PR your agent opened, spot a failing CI job, request a reviewer, and squash-merge — all without leaving the session.

📍 1. Open the workspace rail in your coding-agent session and select Pull Requests. 2. Find the agent's PR in the list, or click `+ (Link a PR)` and paste a GitHub PR URL to add one. 3. On the Summary tab: click the title pencil to rename it, edit the description (paste a screenshot if needed), add a label, and request a reviewer. 4. Hover over the check count to expand individual CI jobs and their run links; ask the agent to fix failures in your next message. 5. When ready, choose `Squash` (or Merge / Rebase) in the Summary tab merge controls. If the agent pushed since you opened the PR, Omnigent will refuse the merge so you can re-review.

— Detailed mechanism and named surfaces, exact UI steps but no CLI command

launch-20261105-26d04a26

02 **Codex thread import into sessions** NEW 80

Codex threads can be imported into Omnigent sessions with full history — messages, tool calls, and tool output become session items, with each Codex compaction recorded as a compaction item; forked or reverted threads carry in history from their base thread. Archived Codex threads import directly into the Archived session list rather than the main list. Sessions are capped at 100,000 items, so threads exceeding that limit or the server's size limit import from their last compaction onward instead of failing; on an HTTP 413 or 'exceeds maximum allowed content length' error, the importer automatically retries from the last compaction and prints a note that only that portion was imported.

— Named limits and error-handling behaviour but no runnable command shown product docs

03 **OpenCode 2.0 required for omnigent opencode** BREAKING 65

`omnigent opencode` now requires OpenCode 2.0 (2.0.18 through 2.0.x); OpenCode 1.x (`opencode-ai`) is no longer supported.

— Exact version range and package name make migration clear v0.21.0

THINNER COVERAGE BELOW

04 **Per-deployment agent and harness selection controls** NEW 55

Deployments can now choose which built-in agents New Chat offers and which one it starts on; individual harnesses and agents can be hidden in Settings › Appearance.

Hide a specific agent harness from the New Chat menu to reduce noise for your team.

> 📍 In the desktop app, go to Settings › Appearance and toggle off the individual harness or agent you want to hide.

— Exact UI path given but limited mechanism detail v0.21.0

05 **Inline GitHub review conversations in Changes view** NEW 50

Users can read, reply to, and start GitHub review conversations inline in the Changes view, with the composer highlighting PRs that need attention.

— Describes behaviour but no exact navigation or command v0.21.0

06 **Plain-language agent tool-call rendering** IMPROVED 50

Agent tool calls in chat now read as plain-language actions with a one-line result and expand to labeled fields instead of raw JSON; file edits show line counts and a per-file diff.

— Clear before/after UI change but no navigation given `v0.21.0`

---

07 **macOS 13 Ventura minimum for desktop app** BREAKING 50

The desktop app now requires macOS 13 Ventura or later; macOS 12 Monterey is no longer supported.

— Exact OS versions named, straightforward upgrade requirement `v0.21.0`

---

08 **Per-side-chat model and reasoning-effort controls** NEW 45

Side chats now have their own model and reasoning-effort controls, letting follow-ups use a different model or effort level than the parent chat.

— Explains the change but no UI path or default given `v0.21.0`

---

09 **Local session integrations for browser and editor** NEW 45

Local sessions can open localhost apps in the desktop browser pane, and the Files panel can open files in VS Code or Cursor.

— Names the tools involved but no exact steps given `v0.21.0`

---

10 **SSH remote sessions from desktop app** NEW 35

Sessions can now run on any remote machine over SSH, launched directly from the desktop app.

— Single-sentence description with no setup or config details `v0.21.0`

---

11 **Searchable, grouped model picker** IMPROVED 25

Long model lists now have a search box and are grouped by model creator.

— Bare description of a minor UI tweak `v0.21.0`

└──▷ **BREAKING ON UPGRADE**

! `omnigent opencode` now requires OpenCode 2.0 (2.0.18 through 2.0.x); OpenCode 1.x (`opencode-ai`) is no longer supported.

! The desktop app now requires macOS 13 Ventura or later; macOS 12 Monterey is no longer supported.

WAS THIS USEFUL? 👍 👎

CrewAI

Feedback ⓘ announced · Nov 5, 2026 BLOG ↗ RANK

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

> CrewAI shipped an experimental `crewai eval` CLI command that benchmarks model and role combinations directly against your existing agent workflow and outputs machine-readable briefs for coding harnesses.

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

01 `crewai eval` CLI for model and role benchmarking NEW

92

The experimental `crewai eval` command pulls the last execution trace and scores it against your existing workflow without requiring a separate fixture library. Model and role overlays let you benchmark combinations — e.g. a frontier model on a planner agent alongside a cheaper open-weight model on a formatter role — across quality, cost, speed, and token usage. It outputs a structured markdown brief formatted for coding harnesses like Claude Code, Codex, and Antigravity to parse, with binary pass/fail exit codes and role-level failure attribution pointing to the exact agent role or Flow step that broke execution.

— Runnable command with full mechanism and named output format described.

launch-20261105-e291cf87

WAS THIS USEFUL? 👍 👎

Arcade

Feedback ⓘ updated · Nov 5, 2026 NOTES ↗ RANK

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

> Arcade added an engine CLI command for safely purging an organization's stored tool execution history, with dry-run defaults and explicit confirmation.

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

01 **Engine CLI command to delete execution history** NEW

72

A new engine CLI command deletes an organization's stored tool execution history. It reports each project's recorded executions and logging policy before acting, runs as a dry run by default, and only marks finished executions for deletion once committed with an explicit row cap and reason.

— Describes behavior and safeguards but names no exact flag syntax. snapshot-20261105

WAS THIS USEFUL? 👍 👎

AG-UI

Feedback ⓘ release/2026-11-05 NOTES ↗ RANK

AG-UI: the Agent-User Interaction Protocol. Bring Agents into Frontend Applications.

> AG-UI's Kotlin SDK (0.5.0) adds reasoning, interrupt, and activity-streaming event types plus multimodal message support and new build targets, while shipping a cluster of breaking changes that tighten payload validation, rework StateManager's JSON-patch internals, and rename the SDK's packaging for AG-UI 1.0 compliance.

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

01 Reasoning events replace THINKING_* telemetry DEPRECATED

85

Adds `REASONING_*` events and `ReasoningTelemetryState` in `kotlin-core` and `kotlin-client` to stream agent reasoning steps, replacing the now-deprecated `THINKING_*` events, `ThinkingTelemetryState`, and `AgentState.thinking`.

— Names exact old and new event types with clear migration path release/2026-11-05

02 Stricter payload validation for AG-UI 1.0 BREAKING

85

Targets AG-UI 1.0 with strict event/input decoding and tightened validation that may reject previously accepted payloads: usage counts are now validated as exact safe JSON integers with decimal/exponent spellings normalized, explicit-null checks are tightened across nested typed models and message/outcome names, `MAX_JSON_DEPTH` is reduced from 512 to 128 (refusing deeply nested SSE payloads), and `StateManager` now rejects deltas it previously accepted: replace/remove of a missing member, add under a missing parent, move into its own child, and add without a value.

— Enumerates every tightened validation rule with before/after values release/2026-11-05

03 Internal RFC 6902 patch implementation in StateManager

80

BREAKING

`kotlin-json-patch` dependency is dropped; `StateManager` now uses an internal RFC 6902 implementation, and rejected deltas report `com.agui.client.state.JsonPatchApplicationException` instead of kotlin-json-patch exception types.

— Names exact exception class and dependency swap release/2026-11-05

04 Injectable Clock in CircuitBreaker NEW

70

`kotlin-tools`' `CircuitBreaker` now accepts an injected `kotlinx.datetime` `Clock` for testable time control; the existing single-argument constructor still defaults to `Clock.System`.

— Names exact constructor behavior and default release/2026-11-05

05 **Agent activity streaming events** NEW 68

Adds `ACTIVITY_SNAPSHOT` and `ACTIVITY_DELTA` events, `ActivityMessage`, and `Role.ACTIVITY` in `kotlin-core` to track agent activity streams; activity messages are no longer echoed back in run inputs.

— Names all new activity surfaces and the behavior change release/2026-11-05

06 **SDK packaging and build configuration changes** BREAKING 68

Group ID changed to `com.ag-ui.community`, removing `com.contextable` and `com.agui` references; `minSdk` lowered to 24; Kotlin pinned to `2.1.20`.

— Names exact group ID and version values for migration release/2026-11-05

07 **Interrupt and resume protocol for agent runs** NEW 65

Adds `Interrupt`, `ResumeStatus`, and `ResumeEntry` types on `RUN_FINISHED` outcomes in the Kotlin SDK, letting clients handle agent interruptions and resumptions via the interrupt protocol.

— Names the new types but gives no usage example release/2026-11-05

08 **Simplified RunFinishedSuccessOutcome serialization** BREAKING 60

`RunFinishedSuccessOutcome` serialization now emits only the type key (`{"type":"success"}`); consumers should verify they tolerate missing extra fields.

— Gives exact before/after JSON shape release/2026-11-05

THINNER COVERAGE BELOW

09 **wasmJs and macosArm64 build targets** NEW 55

Adds `wasmJs` and `macosArm64` targets across `kotlin-core`, `kotlin-client`, and `kotlin-tools`, enabling browser and Apple Silicon builds.

— Names exact targets and affected modules release/2026-11-05

10 **Multimodal content in user messages** NEW 40

Adds `UserMessage` multimodal support and preserves multimodal tool contents across history paths.

— No concrete format or size limit given `release/2026-11-05`

11 **Tool-call error field on ToolMessage** NEW 40

Adds the `ToolMessage.error` field in `kotlin-core` to surface tool-call errors.

— Bare field name with no behavior detail `release/2026-11-05`

└──▷ BREAKING ON UPGRADE

! Targets AG-UI 1.0; strict event/input decoding and tightened validation may reject previously accepted payloads.

! Usage counts validated as exact safe JSON integers; decimal/exponent spellings normalized.

! Explicit-null checks tightened across nested typed models and message/outcome names.

! `THINKING_*` events, `ThinkingTelemetryState`, and `AgentState.thinking` deprecated; migrate to `REASONING_*` and `ReasoningTelemetryState`.

! `kotlin-json-patch` dependency dropped; `StateManager` now uses an internal RFC 6902 implementation, and rejected deltas report `com.agui.client.state.JsonPatchApplicationException` instead of kotlin-json-patch exception types.

! Group ID changed to `com.ag-ui.community`; `com.contextable` and `com.agui` references removed.

! `minSdk` lowered to 24; Kotlin pinned to 2.1.20.

! `RunFinishedSuccessOutcome` serialization now emits only the type key (`{"type":"success"}`); verify consumers are tolerant of missing extra fields.

! `StateManager` now rejects deltas it previously accepted: replace/remove of a missing member, add under a missing parent, move into its own child, and add without a value.

! `MAX_JSON_DEPTH` reduced from 512 to 128; deeply nested SSE payloads are refused.

! Activity messages are no longer echoed back in run inputs.

WAS THIS USEFUL? 👍 👎

platypus

Feedback ⓘ v4.2.0 NOTES ↗ RANK

Self-hosted AI Agents for your whole team — on your infrastructure, your models, around the clock.

> platypus v4.2.0 adds HTTPS serving via Caddy, a fallback model for Trigger runs, trigger-run failure notifications, and a new result stage in the Tool gate, alongside smaller provider-testing and output-handling improvements.

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

01 **Head/tail retention for shellExec output** IMPROVED 50

platypus now retains the head and tail of `shellExec` output instead of truncating it entirely when output is large.

— Names shellExec and the exact behaviour change v4.2.0

02 **Fallback model for Trigger runs** NEW 45

platypus adds a Fallback model for Trigger runs so a secondary model is used automatically when the primary model fails.

— Explains behaviour but no config surface named v4.2.0

03 **Result stage in Tool gate** NEW 45

platypus adds a result stage to the Tool gate, giving tools a dedicated phase to evaluate and act on run outcomes.

— Describes new phase but no concrete API/config shown v4.2.0

04 **Provider model testing in UI** NEW 45

platypus now lets you test a Provider model directly from the provider form, and lets you dismiss a model's test result from the UI.

— Names provider form UI path as starting point v4.2.0

05 **HTTPS serving via Caddy** NEW 35

platypus now serves the Compose stack over HTTPS using Caddy.

— Names Caddy and HTTPS but no config detail v4.2.0

06 **Trigger-run failure notifications** NEW 25

platypus now sends notifications when a trigger run fails.

— Bare description, no notification channel or config named v4.2.0

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.9.15 NOTES ↗ RANK

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

> HyperFrames shipped a new motion-blur timing control, a searchable catalog, and a CLI self-check command that lets agent workflows detect outdated tooling before rendering.

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

01 CLI update check for agents NEW

75

Running `hyperframes update --check` reports whether the CLI or its skills are out of date, surfacing the status to agents automatically so they can prompt for an upgrade before attempting a render.

Let an agent detect a stale CLI or skills set before attempting a render, so it can prompt the user to upgrade rather than fail silently.

```
$ hyperframes update --check
```

— Runnable command with a concrete agent-workflow use case. v0.9.11

02 Motion-blur shutter phase control NEW

64

A new `data-motion-blur-phase` attribute controls when the motion-blur shutter opens within each frame.

— Names the exact attribute but not its value range or default. v0.9.15

THINNER COVERAGE BELOW

03 Catalog search box NEW

48

The catalog now has a search box over the cards that fills the `?q=` filter the gallery already reads.

— Names the query param but gives no UI location detail beyond 'over the cards'. v0.9.14

WAS THIS USEFUL? 👍 👎

LangChain LangGraph

Feedback ⓘ 1.2.20 NOTES ↗ RANK

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

> LangGraph's latest releases focus on precise interrupt handling and SDK ergonomics, adding unique interrupt IDs, a strict resume mode, and finer control over thread state retrieval and run cancellation diagnostics.

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

01 Selective field retrieval for thread state and history NEW

75

The LangGraph SDK adds a `select` parameter to `threads.get_state` and `threads.get_history`, letting callers filter which state keys are returned when retrieving thread state or history.

Retrieve only specific state keys from a thread to reduce payload size when polling long-running agent state.

```python
state = await client.threads.get_state(thread_id="<thread_id>",
select=["messages", "tool_calls"])
```

— Named methods and parameter with a runnable code example sdk==0.4.13

THINNER COVERAGE BELOW

02 Unique interrupt IDs and strict resume mode NEW

53

Each call to interrupt() inside a task now gets its own unique ID, making it possible to target and resume individual interrupts precisely. A new strict resume mode rejects stale interrupt answers — ones that no longer correspond to a live interrupt — rather than silently applying them.

— Explains mechanism but no config flag or API name given 1.2.20

03 Cancel reason on Run objects NEW

35

The LangGraph SDK adds a new `cancel_reason` field on the `Run` object, allowing callers to inspect why a run was cancelled.

— Names the field but gives no usage example or values sdk==0.4.13

WAS THIS USEFUL?  

Vercel AI SDK

Feedback ⓘ ai@7.0.163 NOTES ↗ RANK

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

> Vercel AI SDK's ai@7.0.163 release adds native URL-based audio transcription, provider-owned realtime transports, typed streaming heartbeat parts, word-level transcription timing, and explicit uploaded-video outcomes, alongside telemetry and chat endpoint metadata improvements.

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

01 Native URL transcription for audio

BREAKING 80

Transcribe audio directly from HTTP(S) URLs without downloading and re-uploading — Groq passes the URL through its `url` parameter natively; supply a custom `download` function to retain local download behavior. Because providers may fetch audio natively, `TranscriptionEndEvent.audio.byteLength` and `audio.mediaType` can now be `undefined`, so callers must check for missing values before use.

Transcribe a remote audio file by URL without downloading it first — useful when files exceed Groq's attachment upload limit.

typescript

```typescript
import { transcribe } from 'ai';
import { groq } from '@ai-sdk/groq';

const result = await transcribe({
  model: groq('whisper-large-v3'),
  audio: new URL('https://example.com/interview.mp3'),
});
console.log(result.text);
```

— Names exact param, fallback hook, and breaking field changes with example ai@7.0.163

02 Typed streaming heartbeat ping parts

BREAKING 80

Adds typed streaming heartbeat parts; Anthropic emits them only when `includePings: true` is set (disabled by default), and core forwards pings through stream retries. Consumers with exhaustive stream-part switches must handle the new `ping` union variant when upgrading, even if `includePings` is not enabled.

Handle the new `ping` stream part emitted by Anthropic when `includePings: true` is set, to avoid exhaustive-switch type errors.

```typescript
import { streamText } from 'ai';
import { anthropic } from '@ai-sdk/anthropic';

const { fullStream } = await streamText({
  model: anthropic('claude-opus-5-5', { includePings: true }),
  prompt: 'Explain zero-trust networking.',
});

for await (const part of fullStream) {
  if (part.type === 'ping') {
    // keepalive — no content to process
    continue;
  }
  if (part.type === 'text-delta') {
    process.stdout.write(part.textDelta);
  }
}
```

— Named flag, union variant, and migration example given `ai@7.0.163`

03 **Provider-owned realtime transport support** NEW 70

Adds an optional `createTransport` interface and `provider` connection capability to support provider-owned realtime transports, built on the new experimental `RealtimeModelV4` server-side spec types. Re-exports `Experimental_RealtimeServerConnection`, `Experimental_RealtimeSessionIntent`, and `Experimental_RealtimeUsage` aliases.

— Names interface and exported types but no runnable example `ai@7.0.163`

THINNER COVERAGE BELOW

04 **Word-level transcription timing** IMPROVED 50

Exposes optional transcription word timings separately from segments, sharing `timestampGranularities` across timing settings; OpenAI now sends `timestamp_granularities` only with `verbose_json` and warns when a requested timing is unavailable.

— Names config key and provider behavior but no usage example `ai@7.0.163`

05 **Telemetry content recording controls** IMPROVED 50

Respects `recordInputs` and `recordOutputs` settings when delivering telemetry events to integrations and the `ai:telemetry` channel, filtering nested content and

omitting raw span results when content recording is disabled while preserving application results.

— Names config flags and channel, lacks runnable example `ai@7.0.163`

06 **Uploaded video generation outcome** NEW 45

Represents videos delivered to caller-managed storage with an explicit uploaded outcome — video generation reports `uploadedVideos` without downloading video bytes; callers should access `video` only when `videos` is nonempty.

— Names new field and access pattern, no example shown `ai@7.0.163`

07 **HTTP endpoint metadata on chat telemetry** IMPROVED 45

Reports the actual OpenAI Chat HTTP endpoint (also used by Azure Chat and Bedrock Mantle Chat) through optional request metadata and model-call end events, recording host and port on OpenTelemetry chat spans; exposes only the sanitized HTTP origin in text and object generation results and step/finish callbacks.

— Names affected providers and telemetry fields, no example `ai@7.0.163`

08 **Stream steering messages to UI** NEW 15

Stream steering messages to the UI and keep them in converted messages.

— Only a bare one-line description with no mechanism or names `ai@7.0.163`

└──▷ BREAKING ON UPGRADE

! `TranscriptionEndEvent.audio.byteLength` and `audio.mediaType` can now be `undefined` when a provider fetches audio natively — check for missing values before using them as a number or string.

! Consumers with exhaustive stream-part switches must handle the new `ping` union variant when upgrading, even when `includePings: true` is not set.

WAS THIS USEFUL? 👍 👎

cotal

Feedback ⓘ v0.145.0 NOTES ↗ RANK

The open standard for agent coordination

> Cotal's latest releases focus on tighter Claude Code integration and chat infrastructure: TLS mismatches are now surfaced distinctly in diagnostics, Claude Code seats can wake via a registered MCP entry, channel posts gained @mention waking, and chat composer internals were opened up for custom front ends — alongside several breaking changes to the chat/workspace APIs.

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

01 Claude Code wake channel via MCP registration NEW

95

`cotal update` registers the plugin server as the user-scope MCP entry `cotal` (via `claude mcp add --scope user`) so managed Claude Code seats wake on peer messages; `cotal status` reports this under Claude wake channel. A managed launch with no `cotal` server in the user config or project `.mcp.json` is now refused with a message naming `cotal setup`.

Re-register the user-scope MCP entry on each machine that spawns Claude seats so wake-on-peer-message works again.

```
$ cotal update
```

Check whether the Claude wake channel is correctly registered after updating.

```
$ cotal status
```

— Runnable commands and exact config paths provided. v0.145.0

02 Breaking API changes to workspace chat library BREAKING

90

`openChatSession` now takes the front end's `words` (`ChatWords`) in place of `reopen`; `grantRefusal` and `resolveDmTarget` take `words` too, and `parseRecipient` takes the front end's other way to choose a recipient — all four call sites must be updated. Interfaces that extended `EndpointReply` must be rewritten as type intersections, with TypeScript callers constructing the matching `ok` arm and narrowing `ok` before accessing a required error. `shownTime`,

`concreteChannelProblem` and `offlineRecipientNotice` move from `@cotal-ai/workspace/chat` to the `@cotal-ai/workspace` root, requiring import updates.

— All affected symbols and required migration steps named. v0.142.0

03 TLS-unsupported reporting for TCP listeners BREAKING

75

A TCP listener that does not offer required TLS is now reported as `tls-unsupported` rather than unreachable, with its registration preserved and the transport mismatch shown in raw connection, registration, and `cotal status` diagnostics. Consumers of `ProbeResult` must handle this new failure reason.

— Clear migration note but no code example given. v0.145.0

04 Chat composer rules exported for custom front ends NEW

70

Chat composer rules — typed recipients and their labels, the reply recipient, the Enter rule, slash commands and their outcome words, shared names, the ending notice and the close summary — are now exported from `@cotal-ai/workspace/chat`, letting a custom terminal front end follow the same rules as `cotal chat`.

— Module path named; integration left to reader. v0.142.0

05 Reply-to targeting for chat sessions IMPROVED

65

The shared chat reply rule `replyRecipient` now answers a channel message on its own channel using `replyTo`, and `AdmittedChatSession.send` accepts an optional `replyTo` argument.

— API symbols named, no usage example given. v0.142.0

06 @mention waking in channel posts NEW

60

Channel posts from `cotal chat` now carry `@names` as mentions in the terminal view and `--plain` mode, waking the agents they name — matching behavior already present in `cotal send -c` and the console.

— Named flags but no runnable example shown. v0.142.0

└──▷ BREAKING ON UPGRADE

! Consumers of `ProbeResult` must handle the new `tls-unsupported` failure reason; previously, a TCP listener without required TLS was reported as unreachable.

! `openChatSession` now takes the front end's `words` (`ChatWords`) in place of `reopen`; `grantRefusal` and `resolveDmTarget` take `words` too, and `parseRecipient` takes the front end's other way to choose a recipient — callers must update all four call sites.

! Interfaces that extended `EndpointReply` must be rewritten as type intersections; TypeScript callers must construct the matching `ok` arm and narrow `ok` before accessing a

required error.

! `shownTime`, `concreteChannelProblem` and `offlineRecipientNotice` move from `@cotal-ai/workspace/chat` to the `@cotal-ai/workspace` root — imports must be updated.

WAS THIS USEFUL?  

AWS Strands Agents

Feedback ⓘ typescript/v1.24.0 NOTES ↗

RANK

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

> Strands Agents added a Model.model_info capability-inspection API, a SystemOneDecisionModel integration for low-latency classifiers, S3 vector similarity search for S3Storage, more flexible ClassifierStrategy routing and BeforeInvocationEvent message access in TypeScript, automatic OpenAI prompt caching for GPT-5.6+ models, and smarter Bedrock maxTokens defaults.

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

01 **Model.model_info capability inspection** NEW 80

Inspect model capabilities and limits in Python with the new `Model.model_info` property, which returns structured capability and token-limit metadata — e.g. context window size, output token limit, vision, and tool-use support — so a reader can check what a model supports before wiring up an agent.

Inspect what a Bedrock model supports and its token limits before wiring up an agent.

python

```python
from strands import Agent

agent = Agent()
info = agent.model.model_info
print(info)  # capabilities and output token limit
```

Inspect what a model supports — context window size, output token limit, vision, tool use — before choosing how to configure an agent.

python

```python
from strands import Agent

agent = Agent()
info = agent.model.model_info
print(info)  # structured capabilities and limits, e.g. max
output tokens
```

— Runnable example with exact property access shown harness-cli/v0.2.4

harness-python/v0.3.4

02 SystemOneDecisionModel for Jev and decisionapi.net NEW

75

Adds a `SystemOneDecisionModel` for Jev and decisionapi.net in the Python SDK, intended as a fast, low-latency classifier backend for the agent decision loop; it can be passed to `ClassifierStrategy` or `HumanInTheLoop` as the decision model.

Use decisionapi.net as the decision backend for a classifier in a Python agent.

python

```python
from strands_harness.decision import SystemOneDecisionModel

model = SystemOneDecisionModel()
# pass to ClassifierStrategy or HumanInTheLoop as the decision
model
```

— Named class plus runnable import and usage example `typescript/v1.24.0`

`harness-typescript/v0.3.4` `harness-python/v0.3.4`

03 S3 vector similarity search for S3Storage NEW

75

Adds an `S3VectorSearchStrategy` for `S3Storage` in the Python SDK, enabling vector similarity queries over data stored in S3 (e.g. storage.search(query, top_k=5)).

Use S3 vector similarity search to retrieve semantically relevant chunks stored in S3.

python

```python
from strands.storage import S3Storage
from strands.storage.strategies import S3VectorSearchStrategy

storage = S3Storage(bucket='my-bucket',
strategy=S3VectorSearchStrategy())
results = storage.search('explain agent lifecycle', top_k=5)
```

— Named strategy class with runnable search example `typescript/v1.24.0`

`harness-cli/v0.2.4`

THINNER COVERAGE BELOW

04 **BeforeInvocationEvent message access in TypeScript** NEW

50

Access the full message list from `BeforeInvocationEvent` in TypeScript via the new `messages` field, enabling pre-invocation inspection or mutation of conversation history before an agent turn begins.

— Names the event and field but gives no code example `typescript/v1.24.0` `python/v1.64.0`

05 **Bedrock maxTokens defaults to model limit** IMPROVED

50

Bedrock `maxTokens` now defaults to the model's output limit (Python and TypeScript) instead of requiring an explicit value.

— Clear before/after behaviour but no runnable example `harness-cli/v0.2.4`

06 **ClassifierStrategy supports any DecisionModel** IMPROVED

45

Route with any `DecisionModel` from `ClassifierStrategy` in TypeScript, not just the built-in classifiers.

— Names the surfaces but no example or mechanism detail `typescript/v1.24.0`

07 **OpenAI prompt caching for GPT-5.6+** IMPROVED

45

Strands Agents now places explicit cache points on GPT-5.6+ models by default in the OpenAI provider.

— Names model threshold and provider, no example given `typescript/v1.24.0`

WAS THIS USEFUL?  

LangChain

Feedback ⓘ langchain==1.4.9 NOTES ↗ RANK

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

> LangChain's latest patch releases standardize how refusals from language models are represented, giving a single content block type that works across OpenAI, Anthropic, and Gemini-backed chains.

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

01 Standardized refusal content blocks across providers

BREAKING

47

LangChain now surfaces model refusals (including Gemini safety blocks) as a consistent refusal content block type, letting agents and chains detect and handle declines uniformly regardless of which chat model provider is used. This requires `langchain-core>=1.7.5`, so installations pinned to an older langchain-core will break on upgrading to `langchain==1.4.9`.

— Describes behavior and version requirement but no API names or examples.

`langchain==1.4.9` `langchain-openai==1.7.4` `langchain-anthropic==1.7.10`

└──▷ **BREAKING ON UPGRADE**

! `langchain-core>=1.7.5` is now required; installations pinned to an older version of langchain-core will break on upgrade to langchain==1.4.9.

WAS THIS USEFUL? 👍 👎

claw-orchestrator

Feedback ⓘ v7.20.0 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's latest releases add goal-mode execution so fan-out agents can run full objectives to completion with resumability, alongside per-agent working directories and a new Anthropic-compatible gateway for routing and pricing Claude sessions.

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

01 Anthropic-compatible gateway routing for Claude sessions NEW

91

Route any `claude` session through an Anthropic-compatible gateway by setting `CLAWO_CLAUDE_GATEWAY=1` with the gateway URL in `ANTHROPIC_BASE_URL` or the session's `baseUrl`. claw-orchestrator passes the model name to the CLI as given, skips the built-in proxy, and this applies to all session types including workflow, fan-out, and council nodes.

Point all claude sessions — including workflow, fan-out, and council nodes — at an internal Anthropic-compatible gateway so custom or private model names are passed through without the built-in proxy.

```
$ export CLAWO_CLAUDE_GATEWAY=1
export ANTHROPIC_BASE_URL=https://my-gateway.internal/v1
# then start claw-orchestrator as normal; every claude session
uses the gateway
```

— Env vars, scope, and bypass behaviour fully specified. v7.18.0

02 Goal-mode execution for fan-out agents NEW

85

Set `objective` on a fan-out agent with engine `claude` or `codex` to run it as a goal — judged, budgeted, timed out, and resumable after stalls — instead of a single turn. Each goal runs in a child run checkpointed on the fan-out node, so a resuming parent reattaches to it instead of restarting, and each result carries the goal's run id and final status.

Implement the same project in parallel across two agents on different engines, each running to completion as a goal rather than a single turn.

```yaml
agents:
  - engine: claude
    objective: implement-auth-module
    cwd: ./impl-claude
  - engine: codex
    objective: implement-auth-module
    cwd: ./impl-codex
```

— Names mechanism (checkpoint, reattach) and exact config field. v7.20.0

03 **Per-agent working directories for fan-out nodes** NEW

80

Set a working directory per fan-out agent using the `cwd` field on fan-out nodes and `fanout_start`, either absolute or relative to the fan-out's directory. Each result records the `cwd` it ran in, and an unsafe path fails only that agent rather than the whole run.

Run multiple fan-out agents in isolated directories so each writes code without stepping on the others.

```yaml
fanout_start:
  cwd: ./agents/agent-a

fanout_start:
  cwd: ./agents/agent-b
```

— Exact config field and failure behaviour given. v7.19.0

04 **Gateway model pricing overrides** NEW

75

Price non-Claude gateway models from `pricingOverrides` under `CLAWO_CLAUDE_GATEWAY` instead of borrowing another model's rate; without an override, cost is recorded as 0.

Override pricing for a non-Claude model behind the gateway so costs are tracked accurately instead of defaulting to 0.

```
$ CLAWO_CLAUDE_GATEWAY=1 \
  pricingOverrides='{"my-custom-model": {"input": 0.003,
"output": 0.015}}' \
  claw-orchestrator run
```

— Names the exact env var and default fallback behaviour. v7.19.0

WAS THIS USEFUL? 👍 👎

yoagent

Feedback ⓘ v0.27.0 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.

> yoagent's v0.27.0 release tightens image handling across providers and tools — limiting, filtering and routing images correctly for text-only and OpenAI-compatible models — while adding opt-in sub-agent depth and retry controls and fixing a verifier-bypass and a tool-call panic.

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

01 **Image type and size filtering in read_file** NEW 90

`read_file` now limits images sent to a model to PNG, JPEG and WebP, up to 8000px and 3.75MB, configurable via `with_max_image_bytes` on `ReadFileTool`.

Cap image size sent to a provider to avoid rejected requests from oversized images.

rust

```rust
use yoagent::tools::ReadFileTool;

let tool = ReadFileTool::default()
    .with_max_image_bytes(1_000_000); // 1 MB cap instead of 3.75 MB default
```

— Named config method, exact formats and limits, runnable example v0.27.0

02 **Text-only model image fallback** NEW 75

Setting ModelConfig::with_image_input(false) tells a text-only model to receive a descriptive note in place of each image instead of a broken attachment.

Use a text-only model in a pipeline that may receive image tool results — the model gets a descriptive note instead of a broken image attachment.

rust

```rust
use yoagent::provider::ModelConfig;

let config = ModelConfig::groq("llama-3.1-8b-instant", "Llama 3.1 8B")
    .with_image_input(false);
```

— Named method with runnable example v0.27.0

03 **New public fields on file and bash tool structs**

BREAKING

75

`ReadFileTool`, `WriteFileTool` and `BashTool` gain public fields (`read_log`, `max_image_bytes`, `raw_output`); struct literals that do not use ..Default::default() will fail to compile.

— Exact fields and migration condition named, no code sample v0.27.0

---

04 **Opt-in sub-agent depth and retry controls** NEW

70

New opt-in controls: `SubAgentTool::with_max_depth` bounds how deep a sub-agent can spawn further sub-agents, and `Agent::with_max_retry_after` controls loop retry behaviour.

Limit how deep a sub-agent can spawn further sub-agents, to bound runaway recursion in multi-agent pipelines.

rust

```rust
use yoagent::sub_agent::SubAgentTool;

let tool = SubAgentTool::new(provider.clone(), config.clone())
    .with_max_depth(3);
```

— Depth control has an example; retry control is unexplained v0.27.0

---

05 **ANSI stripping for bash tool output** IMPROVED

65

`bash` tool output now strips ANSI escape sequences by default; calling with_raw_output(true) on `BashTool` keeps the raw sequences.

— Exact flag named but no usage example v0.27.0

---

06 **Tool-result images reach OpenAI-compatible models** IMPROVED

65

Images from tool results now reach OpenAI-compatible models (including DeepSeek) in a user message, fixing a 400 error previously returned by OpenAI and images silently dropped by DeepSeek.

— Named providers and before/after behaviour, no new call site v0.27.0

---

07 **Unanswered tool calls resolved after interrupted runs**

IMPROVED

60

Unanswered tool calls from a failed, cancelled or killed run are now answered at the next run start as not run or outcome unknown, preventing a panic in `continue_loop`.

— Exact bug and mechanism named, no API to invoke v0.27.0

THINNER COVERAGE BELOW

08 Verifier check extended to cut-off answers IMPROVED

50

`on_stop` now also judges answers that were cut off at the output limit, closing a gap where a required verifier could previously be skipped.

— Explains the fixed behaviour but no config surface or example v0.27.0

└──▷ BREAKING ON UPGRADE

! `ReadFileTool`, `WriteFileTool` and `BashTool` gain public fields (`read_log`, `max_image_bytes`, `raw_output`); struct literals that do not use ..Default::default() will fail to compile.

WAS THIS USEFUL? 👍 👎

holmesgpt

Feedback ⓘ 0.46.0 NOTES ↗ RANK

SRE Agent - CNCF Sandbox Project

> HolmesGPT 0.46.0 adds webhook alert destinations, per-MCP-server tool allowlists, and a hosted multi-account AWS MCP toolset, alongside LLM retry handling and token-limit visibility, while removing a field from the chat API.

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

01 **Tool allowlists for MCP servers** NEW 70

Restrict which tools Holmes loads from an MCP server by setting `allowed_tools` in the MCP server configuration, so only the subset you need is exposed to the agent.

— Names exact config key and mechanism, but no example shown 0.46.0

THINNER COVERAGE BELOW

02 **Hosted multi-account AWS MCP toolset** NEW 45

Serve all AWS accounts from a single hosted AWS MCP toolset instead of one per account, reducing deployment overhead.

— Describes benefit but no setup detail or flag 0.46.0

03 **follow_up_actions removed from chat API** BREAKING 45

The `follow_up_actions` field is removed from the chat API, a breaking change for any integration relying on it.

— Names exact field and API but no migration path given 0.46.0

04 **Webhook alert destination** NEW 35

Send alert findings to a webhook destination — a new destination type added alongside the existing Slack integration.

— No config key or endpoint named for the webhook 0.46.0

05 **Token limit visibility in status output** IMPROVED 35

Holmes now reports each model's token limits and their source in the status output, giving operators visibility into which limits are in effect.

— Mentions status output but no exact command or field 0.46.0

06 **Automatic LLM error retries** IMPROVED 35

Retry LLM rate-limit, overload, and 503 errors with backoff automatically instead of failing the run.

— Names error types but no configurable parameters `0.46.0`

└──▷ **BREAKING ON UPGRADE**

! The `follow_up_actions` field is removed from the chat API.

WAS THIS USEFUL? 👍 👎

PydanticAI

Feedback ⓘ v2.66.0 NOTES ↗ RANK

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

> PydanticAI's v2.66.0 release adds self-hosted and Render sandbox environments, Azure Durable Task durability and Pixeltable memory to its harness, finer control over MCP instruction injection and compaction retries, and reports xAI Grok Voice timing, alongside three breaking behaviour changes around Anthropic web tools, tool approval gates, and compaction message matching.

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

01 MCP instruction injection control NEW

75

Control whether MCP server instructions are injected into the agent's system prompt with `include_instructions` on the `MCP` capability; set it to `False` to exclude the server's default instructions when you manage system prompts yourself.

Exclude MCP server instructions from the agent prompt when you manage system instructions yourself and do not want the server's defaults mixed in.

python

```python
from pydantic_ai import Agent
from pydantic_ai.capabilities import MCP

agent = Agent(
    'openai:gpt-6.1-sol',
    capabilities=[
        MCP(server_url='https://my-mcp-server.example.com',
include_instructions=False),
    ],
)
```

— Names exact flag and capability with a runnable example. v2.66.0

02 Retry configuration for context compaction NEW

70

Add a `retries` field to `SummarizingCompaction` to configure how many times context compaction is retried before failing, allowing recovery from transient model failures during long-running sessions.

Allow compaction to retry on transient model failures instead of surfacing an error immediately in long-running sessions.

```python
from pydantic_ai_harness import SummarizingCompaction

compaction = SummarizingCompaction(retries=3)
```

— Names field and class with usage example, but limited mechanism detail. v2.66.0

03 **New sandbox, durability and memory integrations in Harness**

NEW

70

Add `OpenSandbox` for running agent workspaces in self-hosted sandbox environments and `RenderSandbox` to run agent workspaces in Render sandboxes; add Azure Durable Task durability support to `pydantic_ai_harness`; and add `Pixeltable` tools and memory storage to Harness.

Run an agent workspace in a self-hosted sandbox rather than a managed cloud provider.

```python
from pydantic_ai import Agent
from pydantic_ai_harness import OpenSandbox

agent = Agent(
    'anthropic:claude-fable-5-1',
    capabilities=[OpenSandbox()],
)
```

— Names four distinct surfaces with one runnable example for one of them. v2.66.0

04 **FunctionToolset approval gate scope expanded**

BREAKING

60

FunctionToolset(requires_approval=True) now also applies to `Tool` instances passed directly to the toolset, which may trigger approval gates that did not fire before.

— Names exact config and behaviour change, no example given. v2.66.0

THINNER COVERAGE BELOW

05 **xAI Grok Voice segment timestamps** NEW

55

Report xAI Grok Voice speech timing as `segment` timestamps via `output_speech_timestamps`.

— Names the flag and format but no usage example. v2.66.0

06 **Anthropic web tools citation behaviour change**

BREAKING

55

Calling Anthropic's `20260209` web tools directly on the Anthropic API when citations are requested changes behaviour from the previous approach; verify citation-dependent workflows after upgrading.

— Describes the change and gives a migration warning, no code surface. v2.66.0

---

07 **SummarizingCompaction kept-message matching changed** BREAKING

55

SummarizingCompaction(keep_user_messages=True) now identifies kept-message copies only by their text marker, so copies that previously matched by other means may no longer be recognised.

— Names config and mechanism change but no remediation example. v2.66.0

---

08 **Workspace authority for uncontrolled files** NEW

40

Add `Workspace.authority` and `AuthorityWorkspace` for workspaces whose files the operator does not control.

— Names the classes but gives no mechanism or example. v2.66.0

└──▷ **BREAKING ON UPGRADE**

! Calling Anthropic's `20260209` web tools directly on the Anthropic API when citations are requested changes behaviour from the previous approach; verify citation-dependent workflows after upgrade.

! FunctionToolset(requires_approval=True) now also applies to `Tool` instances passed directly to the toolset, which may trigger approval gates that did not fire before.

! SummarizingCompaction(keep_user_messages=True) now identifies kept-message copies only by their text marker, so copies that previously matched by other means may no longer be recognised.

WAS THIS USEFUL? 👍 👎

Mastra

Feedback ⓘ @mastra/core@1.82.0 NOTES ↗ RANK

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

> Mastra ships a full Knowledge page-tree system — agent CRUD tools, server-side sync with vector/hybrid search, and opt-in HTTP routes with an SSE stream — alongside durable worker continuity across restarts, richer trace filtering and latency aggregates, guardrail audit logging, and several breaking removals of experimental APIs.

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

01 **Agent CRUD tools for Knowledge page tree** NEW 80

Mastra adds a full toolset for agents to manage the Knowledge page tree: `knowledge_list` to browse the tree, `knowledge_read` for line/section/grep-style reads, `knowledge_page` (`op: 'write' | 'move' | 'merge'`), `knowledge_note` (`op: 'add' | 'restamp' | 'remove'`), and `knowledge_pin` (`op: 'pin' | 'unpin'`), giving an agent full CRUD over pages, notes, and pins.

Attach Knowledge with full read/write tools and pinned context to an agent, so it can browse, search, and edit a scoped page tree each turn.

typescript

```typescript
import { Agent } from '@mastra/core';
import { Knowledge } from '@mastra/core/knowledge';
import { knowledgeMemoryScopes } from '@mastra/memory';
import { PostgresStore } from '@mastra/pg';

const knowledge = new Knowledge({
  storage: new PostgresStore({ id: 'kb', connectionString:
process.env.DATABASE_URL }),
  scopes: { ...knowledgeMemoryScopes() },
  tools: { read: true },
  pinnedContext: { maxPins: 5, maxTokens: 2000 },
});

const agent = new Agent({
  id: 'research',
  name: 'Research Agent',
  instructions: 'Answer questions using the knowledge base.',
  model,
  knowledge,
  scopes: ({ scopes }) => scopes,
});
```

— Names every tool and op verbatim but lacks a direct call example `@mastra/core@1.82.0`

02 **Advanced trace query filters** NEW 80

Filter advanced trace queries by `requestId`, `serviceName`, and `experimentId` using `eq`, `ne`, `in`, `notIn`, `exists`, or `notExists`, applied at trace scope or inside `spans.some` / `spans.none`.

Filter trace queries by request ID and service name to isolate spans from a specific payment-service call when debugging a production incident.

typescript

```
const result = await mastraClient.queryTraces({
  timeRange: { from: '2026-11-01T00:00:00.000Z', to: '2026-11-
02T00:00:00.000Z' },
  where: {
    op: 'and',
    args: [
      { op: 'eq', left: { path: 'requestId' }, right: { literal:
'req-42' } },
      { spans: { some: { op: 'eq', left: { path: 'serviceName' },
right: { literal: 'payments' } } } },
    ],
  },
});
```

— Runnable queryTraces example with exact filter syntax `@mastra/core@1.82.0`

03

Audit logging for classifier guardrail decisions NEW

80

Add `onDecision` to `ClassifierProcessor` to write guardrail decisions — outcome (`allowed`, `filtered`, or `aborted`), answers or error, call duration, and message/thread/resource/run IDs — to an audit log after each classification.

Add `onDecision` to a `ClassifierProcessor` to record every guardrail outcome — including classifier failures — in an audit log.

typescript

```
import { ClassifierProcessor } from '@mastra/core';

new ClassifierProcessor({
  classifier: safetyClassifier,
  onResult: (answers, { abort }) => {
    if ((answers.unsafe.probability ?? 1) > 0.8) {
      abort('Blocked', { metadata: { policy: 'unsafe' } });
    }
  },
  onDecision: async decision => {
    await auditLog.insert({
      threadId: decision.ids.threadId,
      outcome: decision.outcome,
      answers: decision.result ? decision.result.answers : null,
      error: decision.result ? null : String(decision.error),
    });
  },
});
```

— Full runnable example with every logged field named `@mastra/core@1.82.0`

04 **Read-only Knowledge HTTP routes and client APIs** NEW

70

Enable opt-in read-only Knowledge HTTP routes (tree/page/graph/search/activity) on Mastra Server, plus an SSE change-event stream, with matching listKnowledge() and getKnowledge() APIs in `@mastra/client-js`.

— Routes and client methods named, but no exact endpoint paths given `@mastra/core@1.82.0`

05 **Temporal steps validate input before running** BREAKING

70

`@mastra/temporal@0.5.0` now validates `inputSchema` and `requestContextSchema` before running steps, which can fail runs that previously succeeded with wrong input; set `validateInputs: false` to opt out.

— Exact opt-out flag and behavior change named `@mastra/core@1.82.0`

06 **Server-side Knowledge content sync** NEW

65

Sync external content directly from server code using knowledge.upsertPage(), knowledge.removePage(), and note upserts, with optional vector/hybrid search and chunked embeddings for long pages.

— Named methods and mechanism but no code sample given `@mastra/core@1.82.0`

THINNER COVERAGE BELOW

07 **Per-message scope narrowing for agents** NEW

55

Let an agent's `scopes` function narrow or replace a run's incoming scopes, with session.sendMessage({ content, scopes }) support so a message can carry its own scope set.

Attach Knowledge with full read/write tools and pinned context to an agent, so it can browse, search, and edit a scoped page tree each turn.

```typescript
import { Agent } from '@mastra/core';
import { Knowledge } from '@mastra/core/knowledge';
import { knowledgeMemoryScopes } from '@mastra/memory';
import { PostgresStore } from '@mastra/pg';

const knowledge = new Knowledge({
  storage: new PostgresStore({ id: 'kb', connectionString:
process.env.DATABASE_URL }),
  scopes: { ...knowledgeMemoryScopes() },
  tools: { read: true },
  pinnedContext: { maxPins: 5, maxTokens: 2000 },
});

const agent = new Agent({
  id: 'research',
  name: 'Research Agent',
  instructions: 'Answer questions using the knowledge base.',
  model,
  knowledge,
  scopes: ({ scopes }) => scopes,
});
```

— Mechanism and API named, example shows partial usage `@mastra/core@1.82.0`

08 **Worker continuity across process restarts** NEW 55

Enable worker continuity across API process restarts by configuring a fencing-capable shared storage adapter such as `PostgresStore` from `@mastra/pg` together with a shared pubsub transport.

— Names adapter and transport but no setup example `@mastra/core@1.82.0`

09 **Async workflow step-result access** DEPRECATED 50

Add `getStepResultAsync` to workflow step, condition, sleep, and .map() contexts as a promise-returning replacement for the now-deprecated `getStepResult`.

— Names replacement and deprecated API but no example `@mastra/core@1.82.0`

10 **Removal of experimental subconscious and Memory knowledge option** 50

BREAKING

`experimental_subconscious` and the Memory `knowledge` option are removed and now throw; Knowledge must be attached to the Agent instead, and older `Subconscious` and `KnowledgeSemanticIndexCoordinator` constructors now throw.

– Names removed APIs but no migration code shown `@mastra/core@1.82.0`

11 **Removal of knowledge.reindex()** BREAKING 50

knowledge.reindex() is removed; use upsertPage(), note writes, and reindexLinks() for link indexing instead.

– Clear replacement methods named, no example `@mastra/core@1.82.0`

12 **Streaming latency fields in trace aggregates** NEW 45

Trace and span aggregates now expose `timeToFirstToken.*` and `timeToFirstChunk.*` fields so dashboards can chart streaming latency, gated by store capability.

– Named fields but no usage example or exact field list `@mastra/core@1.82.0`

13 **Removal of factory work-item claiming APIs** BREAKING 30

`@mastra/factory@0.27.0` removes work-item claiming APIs: `claimKey` and related methods and types are removed.

– Bare removal notice with no replacement guidance `@mastra/core@1.82.0`

└──▷ **BREAKING ON UPGRADE**

! `@mastra/temporal@0.5.0`: Steps now validate `inputSchema` and `requestContextSchema` before running, which can fail runs that previously succeeded with wrong input; set `validateInputs: false` to opt out.

! `experimental_subconscious` and the Memory `knowledge` option are removed and now throw; Knowledge must be attached to the Agent, and older `Subconscious` and `KnowledgeSemanticIndexCoordinator` constructors now throw.

! knowledge.reindex() is removed; use upsertPage(), note writes, and reindexLinks() for link indexing instead.

! `@mastra/factory@0.27.0` removes work-item claiming APIs: `claimKey` and related methods and types are removed.

WAS THIS USEFUL?  

ag2

Feedback ⓘ v1.3.0 NOTES ↗ RANK

AG2 (formerly AutoGen): The Open-Source AgentOS.Join us at: https://discord.gg/sNGSwQME3x

> ag2 1.3.0 adds delegation-tree observability and configurable context injection to its agent runtime, alongside a composable KnowledgePlugin and history query API, while deprecating the task framework, assembly parameter and OpenAIConfig ahead of a 1.4.0 removal and raising the minimum Python version to 3.11.

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

01 **Explicit variable scoping across delegation** NEW 85

Define variable scope across delegation boundaries with `ScopedValue` and `Scope.SHARED`, `Scope.LOCAL`, or `Scope.COPIED`, controlling whether a variable is shared between coordinator and workers, kept local, or copied.

Share a findings list across a coordinator and all its workers so every delegated call appends to the same collection, while keeping the coordinator's draft private.

```python
from ag2 import Agent, Context, Scope, ScopedValue, observer,
tool
from ag2.config import OpenAIResponsesConfig
from ag2.events import ChildEvent, ModelMessageChunk

config = OpenAIResponsesConfig(model='gpt-5', streaming=True)

@tool
def save_finding(finding: str, context: Context) -> str:
    context.variables['findings'].append(finding)
    context.variables['draft'].append(finding)
    return 'Saved'

@observer(ChildEvent)
def show_progress(event: ChildEvent) -> None:
    source = event.source_event
    if isinstance(source, ModelMessageChunk):
        print(f'{event.agent_name}: {source.content}')

worker = Agent('worker', config=config,
    prompt='Call save_finding once to record the requested
finding.',
    tools=[save_finding])

coordinator = Agent('coordinator', config=config,
    prompt='Delegate the task to task_worker once, then summarize
its response.',
    tools=[worker.as_tool(description='Record a finding.')],
    observers=[show_progress],
    variables={
        'findings': ScopedValue([], scope=Scope.SHARED),
        'draft': [],
        'root_note': ScopedValue('Coordinator only',
scope=Scope.LOCAL),
    })

reply = await coordinator.ask('Record that each delegated
invocation has its own conversation history.')
print(reply.context.variables['findings'])  # written by worker
print(reply.context.variables['draft'])     # []
```

— Named scope values with a worked coordinator/worker example. v1.3.0

02 **Typed history queries on conversation stream** NEW 85

Query retained history by event type with stream.history.since_last_user_input().where(...), using .count(), .last(), and .first() selectors to retrieve or audit events from the current turn.

Count and retrieve tool calls made in the most recent user-input interval, useful for auditing what an agent did in a single turn.

```python
from ag2.events import ToolCallEvent

calls =
reply.context.stream.history.since_last_user_input().where(ToolCa
llEvent)
print(await calls.count())
print(await calls.last())
```

— Exact query chain and selectors with runnable example. v1.3.0

---

03 **Dynamic context injection via statusbar_middleware** NEW

80

Inject dynamic context before each model call using `statusbar_middleware`; set `persist=True` to retain the injected snapshots as service observations.

Inject a live status bar into every model call so the agent always sees the current goal and remaining steps before deciding what to do next.

```python
from ag2 import Agent, Context, DataInput
from ag2.config import OpenAIResponsesConfig
from ag2.middleware import statusbar_middleware

config = OpenAIResponsesConfig(model='gpt-5', streaming=True)

@statusbar_middleware(persist=True)
def task_status(context: Context) -> DataInput:
    return DataInput({
        'goal': context.variables['goal'],
        'remaining_steps': context.variables['remaining_steps'],
    })

agent = Agent(
    'writer',
    config=config,
    middleware=[task_status],
    variables={
        'goal': 'Prepare release notes',
        'remaining_steps': ['Draft', 'Review'],
    },
)
reply = await agent.ask('Draft the release notes.')
```

— Full mechanism plus runnable decorator example. v1.3.0

04 **Task framework deprecated in favor of Agent.as_tool()**

DEPRECATED 80

`TaskConfig`, Agent(tasks=...), `Agent.tasks`, and the auto-injected `run_subtask` / `run_subtasks` tools are deprecated with warnings and will be removed in 1.4.0 — migrate to Agent.as_tool(). Similarly, Agent.task() is deprecated and will be removed in 1.4.0 — construct Task(owner_id=..., spec=TaskSpec(...)) directly instead.

— Names every deprecated surface and both migration targets. v1.3.0

05 **Agent assembly parameter deprecated for middleware** DEPRECATED

80

Agent(assembly=...) is deprecated and will be removed in 1.4.0 — register `AssemblerMiddleware` via `middleware=` and add `HaltCheckMiddleware` when using `AlertPolicy`.

— Exact replacement classes and parameter named for migration. v1.3.0

06 **ChildEvent observations across delegation tree** NEW

Ancestors now receive typed `ChildEvent` observations from every descendant — including streamed chunks — without those events entering the ancestor's model transcript, enabling progress observers like @observer(ChildEvent) on a coordinator agent.

Share a findings list across a coordinator and all its workers so every delegated call appends to the same collection, while keeping the coordinator's draft private.

python

```python
from ag2 import Agent, Context, Scope, ScopedValue, observer,
tool
from ag2.config import OpenAIResponsesConfig
from ag2.events import ChildEvent, ModelMessageChunk

config = OpenAIResponsesConfig(model='gpt-5', streaming=True)

@tool
def save_finding(finding: str, context: Context) -> str:
    context.variables['findings'].append(finding)
    context.variables['draft'].append(finding)
    return 'Saved'

@observer(ChildEvent)
def show_progress(event: ChildEvent) -> None:
    source = event.source_event
    if isinstance(source, ModelMessageChunk):
        print(f'{event.agent_name}: {source.content}')

worker = Agent('worker', config=config,
    prompt='Call save_finding once to record the requested
finding.',
    tools=[save_finding])

coordinator = Agent('coordinator', config=config,
    prompt='Delegate the task to task_worker once, then summarize
its response.',
    tools=[worker.as_tool(description='Record a finding.')],
    observers=[show_progress],
    variables={
        'findings': ScopedValue([], scope=Scope.SHARED),
        'draft': [],
        'root_note': ScopedValue('Coordinator only',
scope=Scope.LOCAL),
    })

reply = await coordinator.ask('Record that each delegated
invocation has its own conversation history.')
print(reply.context.variables['findings'])  # written by worker
print(reply.context.variables['draft'])     # []
```

— Clear before/after behavior with observer code example. v1.3.0

07 Real-time UsageEvent propagation to ancestors BREAKING

75

Original `UsageEvent` records from descendants now reach ancestors in real time, preserving source identity, stage, model, and provider. As part of this change, new delegations no longer generate subtask rollup `UsageEvent` records with `kind='subtask'`; consumers that grouped spend under that kind must switch to reading source records and their delegation provenance.

— Explains mechanism and migration need but gives no code sample. v1.3.0

---

08 **KnowledgePlugin for composable knowledge retrieval** NEW

70

Configure knowledge retrieval as a composable plugin with KnowledgePlugin(store=...) passed to Agent(plugins=[...]). The prior Agent(knowledge=...) parameter is deprecated and will be removed in 1.4.0 in favor of this plugin interface.

— Names the new plugin API and its deprecated predecessor, no runnable example. v1.3.0

---

09 **Task lifecycle events with capability and TTL** NEW

60

Enable task lifecycle events on any `ask`, `run`, or `resume` call using the `capability` and `ttl_seconds` parameters.

— Names the parameters but no usage example given. v1.3.0

---

10 **OpenAIConfig to become alias in 1.4.0** DEPRECATED

60

`OpenAIConfig` will become an alias for `OpenAIResponsesConfig` in 1.4.0 — use `OpenAIResponsesConfig` or `OpenAICompletionsConfig` explicitly to avoid ambiguity.

— Names the config classes involved but no code example. v1.3.0

THINNER COVERAGE BELOW

11 **Configurable max SSE event size for MCP tools** NEW

50

Configure the maximum accepted SSE event size for MCP tool results with `max_sse_event_size`.

— Named config value but no usage context or example. v1.3.0

---

12 **Gemini response resumption control** NEW

50

Control Gemini response resumption at the per-request token limit with `automatic_continuation`.

— Named flag with minimal mechanism detail. v1.3.0

---

13 **TrackingConfig for complete model input capture** NEW

45

Capture the complete ordered model input for each call with `TrackingConfig`.

— Bare name and one-line description, no example. v1.3.0

14 **Python 3.11 minimum requirement** BREAKING 40

AG2 1.3 requires Python 3.11 or newer; environments still on Python 3.10 must upgrade Python before updating.

— Clear version requirement, no further migration detail. v1.3.0

└──▷ **BREAKING ON UPGRADE**

! AG2 1.3 requires Python 3.11 or newer; environments still on Python 3.10 must upgrade Python before updating.

! `TaskConfig`, Agent(tasks=...), `Agent.tasks`, and the auto-injected `run_subtask` / `run_subtasks` tools are deprecated with warnings and will be removed in 1.4.0 — migrate to Agent.as_tool().

! Agent(assembly=...) is deprecated and will be removed in 1.4.0 — register `AssemblerMiddleware` via `middleware=` and add `HaltCheckMiddleware` when using `AlertPolicy`.

! Agent(knowledge=...) is deprecated and will be removed in 1.4.0 — use `plugins=[KnowledgePlugin(...)]` instead.

! Agent.task() is deprecated and will be removed in 1.4.0 — construct Task(owner_id=..., spec=TaskSpec(...)) directly.

! `OpenAIConfig` will become an alias for `OpenAIResponsesConfig` in 1.4.0 — use `OpenAIResponsesConfig` or `OpenAICompletionsConfig` explicitly.

! New delegations no longer generate subtask rollup `UsageEvent` records with `kind='subtask'` — consumers that grouped spend under that kind must switch to source records and their delegation provenance.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

kordoc

Feedback ⓘ v4.24.46 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 improved document fidelity this window, adding source-cell traceability for spreadsheet tables and better recognition of circled Hangul/Latin characters in OCR and heading parsing.

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

01 Circled Hangul and Latin character recognition IMPROVED

70

kordoc now recognizes circled Hangul jamo and Latin characters (㉠–㉭, ⓐ–ⓩ / ㉮–㉯) that were previously misread as plain characters, dropped entirely, or substituted with circled numerals, by re-reading ring interiors and distinguishing isolated jamo (ㄱ–ㅁ) by stroke position, raising OCR and heading accuracy across Korean document types.

— Explains mechanism and character ranges but no runnable example v4.24.43

02 Source cell addresses in table IR NEW

60

The structured table IR gains `IRTable.sourceRows` and `sourceCols` fields, exposing the original sheet row and column addresses for cells parsed from XLSX and XLS files.

— Names exact fields but no usage example v4.24.46

WAS THIS USEFUL? 👍 👎

godot-mcp

Feedback ⓘ v0.35.0 NOTES ↗ RANK

Godot-MCP — Model Context Protocol (MCP) integration for the Godot Engine. dev.

> Godot-MCP v0.35.0 adds explicit MCP annotations to every tool and surfaces an RQL documentation link in tool metadata for agents.

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

01 **MCP annotations and RQL docs link on tools** IMPROVED

40

Every MCP tool now carries explicit MCP annotations, and tool metadata surfaces an RQL docs link so agents can discover query reference documentation directly from the tool definitions.

— Names the mechanism but no concrete flag or endpoint given v0.35.0

WAS THIS USEFUL? 👍 👎

uml-mcp

Feedback ⓘ v1.4.0 NOTES ↗ RANK

UML-MCP Server is a UML diagram generation tool based on MCP (Model Context Protocol), which can help users generate various types of UML diagrams through natural language description or directly writing PlantUML and Mermaid and Kroki https://uml-mcp.vercel.app/mcp

> uml-mcp v1.4.0 adds a canonical HttpAgent-compatible interface, a shadcn-based admin console with setup wizard and Kroki playground, a D3 diagrams plugin, a Skills extension for agent workflows, YAML-driven audit trail and rate limiting, and cuts MCP connect-time token overhead by roughly a fifth.

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

01 **Admin console with setup wizard and Kroki playground** NEW 65

A new admin console, built on shadcn, ships with a first-run setup wizard available via both web and CLI, writable settings, and a Kroki playground. Users sign in to the hosted console with email and password authentication.

– Names console components and UI entry points but no exact paths v1.4.0

02 **uml-mcp.yaml config with audit trail and rate limiting** NEW 62

Server configuration now lives in a `uml-mcp.yaml` file, which also drives MXCP-style audit trail and rate limiting.

– Names config file and mechanisms but not exact keys v1.4.0

THINNER COVERAGE BELOW

03 **Reduced MCP connect-time token overhead** IMPROVED 50

uml-mcp cuts MCP connect-time token surface from ~5,045 to ~4,118 tokens, reducing context overhead for every connection.

– Gives exact numbers but no action for reader to take v1.4.0

04 **HttpAgent-compatible interface** NEW 45

uml-mcp introduces a new canonical HttpAgent-compatible interface, allowing HttpAgent-compatible clients to connect to the server.

– Names the interface but no connection details or example v1.4.0

05 **D3 diagrams plugin** NEW 45

A new D3 diagrams plugin extends diagram output capabilities, producing HTML output.

— Names plugin and output format, no further mechanism v1.4.0

06 **Skills extension for agent workflows** NEW 39

A new Skills extension adds diagram-rendering capabilities to agent workflows.

— Names extension but lacks usage specifics v1.4.0

07 **Batch gallery and diagram source editing** NEW 37

The apps layer now includes a batch gallery and the ability to edit diagram source directly within it.

— Describes feature but no UI path or command given v1.4.0

WAS THIS USEFUL? 👍 👎

davinci-resolve-mcp

Feedback ⓘ v4.19.0 NOTES ↗ RANK

davinci-resolve-mcp is an MCP server integration for DaVinci Resolve Studio.

> davinci-resolve-mcp's v4.19.0 release expands `drx generate` with offline variable-speed ramp writing, keyframed grade parameters, Magic Mask 2 click authoring, and `lumMixSlider` support.

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

01 **Variable-speed ramp writing in drx generate** NEW 95

Write variable-speed ramps with eased speed points into a closed project's `Project.db` using `list_video_items` and `set_speed_ramp` on `Sm2TiItem` columns `MediaTimemapBA` and Duration; `easeFrames` controls easing, and `dryRun` reports `next.start`, `next.overlapFrames`, and `wouldRefuse` before committing.

— Exact fields, columns, and dry-run output named v4.19.0

02 **Magic Mask 2 click authoring** NEW 83

Place Magic Mask 2 subject clicks on another clip's frame and point with `placeMagicMaskClick`, optionally copying a Primary into the mask node, then track with CreateMagicMask("BI") after `ApplyGradeFromDRX`.

— Named function calls and sequence given, no limits noted v4.19.0

03 **Keyframed grade parameters via keyframeCorrector** NEW 80

Author keyframed grade parameters via `keyframeCorrector`, writing one block per keyframe in a corrector's `F6`; apply with `gradeMode 2` to preserve keyframes or `gradeMode 0` to drop them.

— Concrete flags and mode values given, no broader scope v4.19.0

THINNER COVERAGE BELOW

04 **lumMixSlider support in drx generate** NEW 20

`drx generate` adds support for `lumMixSlider`.

— Named only in title, no further detail given v4.19.0

WAS THIS USEFUL? 👍 👎

slack-skills-plugin

Feedback ⓘ v1.7.0 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 v1.7.0 adds Codex marketplace installation and tightens the reliability of skill hand-offs in the slack-cli skill.

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

01 **Codex marketplace for plugin installs** NEW 60

slack-skills-plugin adds a Codex marketplace so practitioners can install each release with `/plugin marketplace add slackapi/slack-skills-plugin`.

— Names exact install command but no further mechanism given v1.7.0

THINNER COVERAGE BELOW

02 **Skill hand-off reliability fixes in slack-cli skill** IMPROVED 55

Skills that hand off to the `slack-cli` skill now ask the agent to load that skill before running any command, preventing PATH-collision false positives where a different tool named `slack` is found first. The `slack-cli` skill now tells the agent to suggest `slack upgrade` when a command or flag it describes is missing from the installed CLI. Skills now reference each other by bare name (for example, `slack-cli`) instead of the `slack:slack-cli` prefix that only Claude Code registers, so hand-offs work on every install path including `npx skills`.

— Explains mechanism and names surfaces but fixes are internal agent behaviour, not user actions

v1.7.0

WAS THIS USEFUL? 👍 👎

Model Context Protocol Servers

Feedback ⓘ v2.3.0

NOTES ↗ RANK

MCP Servers provides reference server implementations for connecting AI applications to external tools and data.

> MCP Servers v2.3.0 hardens the filesystem and fetch reference servers with new access controls, adds merge and remote tooling to the git server, and improves memory server durability.

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

01 Filesystem server access control and new tools NEW

90

The filesystem server adds a `--read-only` flag that serves only read tools, a `--confine-roots` flag to keep client Roots inside the command-line directories (logging the effective allowed directories and their source), a `file://` URI template that exposes files as MCP resources, an optional depth argument to limit `directory_tree` output, and `append_file` and `copy_file` tools for appending content to files and copying files.

Run the filesystem server in read-only mode so an LLM client can browse files but cannot write, copy, or delete anything.

```
$ npx -y @modelcontextprotocol/server-filesystem --read-only
/path/to/allowed/files
```

— Six named flags/tools with a runnable example command v2.3.0

02 Fetch server host allowlist and timeout NEW

75

`mcp-server-fetch` gains a `--allowed-hosts` operator allowlist to block outbound fetch requests to unlisted hosts, and its HTTP request timeout is now configurable.

Restrict the fetch server to a known set of hosts, preventing an LLM from reaching arbitrary URLs in your environment.

```
$ uvx mcp-server-fetch --allowed-hosts api.example.com
docs.example.com
```

— Named flag with exact command, timeout mechanism unspecified v2.3.0

03 Memory server durability and read modes IMPROVED

70

`server-memory` now warns when the memory file is inside `node_modules`, supports configuring a durable path via the `MEMORY_FILE_PATH` environment variable, and adds lightweight read modes to its read tools.

Store the memory graph in a stable location outside node_modules so it survives package reinstalls.

```
$ MEMORY_FILE_PATH=~/.config/mcp/memory.json npx -y
@modelcontextprotocol/server-memory
```

— Named env var with example, read modes left vague v2.3.0

THINNER COVERAGE BELOW

04 **Git merge, remote and change-prompt tools** NEW 55

`mcp-server-git` adds a `git_merge` tool for merging branches, a read-only `git_remote` tool that lists a repository's remotes, and prompts that embed a repository's changes.

— Tool names given but no usage example v2.3.0

05 **get_time_ago tool in time server** NEW 35

`mcp-server-time` adds a `get_time_ago` tool for expressing past times in relative form.

— Bare tool name, no example or mechanism v2.3.0

WAS THIS USEFUL? 👍 👎

D4Vinci Scrapling

Feedback ⓘ v0.5.1 NOTES ↗ RANK

A Web Scraping framework that handles everything from a single request to a full-scale crawl! Don't be shy, join here: https://discord.gg/EMgGbDceNQ and follow here for daily tips and tricks: https://x.com/Scrapling_dev

> Scrapling v0.5.1 adds CDP-based remote browser sessions for reusing authenticated logins, configurable log file encoding, and two new MCP server capabilities for AI agents to capture page motion and perform drag-and-drop interactions.

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

01 **RemoteSession CDP browser connections** BREAKING 85

Connect to an existing Chromium browser via CDP using the new `RemoteSession` and `AsyncRemoteSession` classes, with `cdp_url` accepting an HTTP or WebSocket endpoint and `new_context=False` reusing the browser's existing login state. Connecting to remote browsers now requires these new classes; existing CDP connection code must be migrated per the remote-session migration guide.

Scrape a site that requires an active login by reusing an already-authenticated Chromium browser instance over CDP.

python

```python
from scrapling.fetchers import RemoteSession

with RemoteSession(cdp_url="http://localhost:9222",
new_context=False) as session:
    page = session.fetch("https://example.com/dashboard")
    print(page.css('h1::text').get())
```

— Named classes and params with runnable example, but migration guide not detailed v0.5.1

THINNER COVERAGE BELOW

02 **Spider log file encoding control** NEW 55

Control spider log file encoding with the new `log_file_encoding` setting, accepting any Python encoding name (e.g. `'utf-8'`); defaults to `None` to preserve Python's default.

— Named config key and default, but no example shown v0.5.1

03 **MCP server drag-and-drop interactions** NEW 50

AI agents using the MCP server can now perform drag-and-drop interactions: hold left, right, or middle mouse buttons, move elements or sliders, inspect the page while holding, then release.

— Mechanism described step by step but no concrete invocation shown v0.5.1

04 **MCP server page motion recording** NEW 40

AI agents using the MCP server can now record page motion as a sequence of timestamped frames to capture animations and changing controls that a single screenshot misses.

— Describes behavior but no command, flag, or example given v0.5.1

└──▷ **BREAKING ON UPGRADE**

! Connecting to remote browsers now requires the new `RemoteSession` or `AsyncRemoteSession` classes; existing CDP connection code must be migrated per the remote-session migration guide.

WAS THIS USEFUL? 👍 👎

kubernetes-mcp-server

Feedback ⓘ v0.1.2 NOTES ↗ RANK

Model Context Protocol (MCP) server for Kubernetes and OpenShift

> kubernetes-mcp-server adds a new tool for discovering CRD apiVersions across a cluster.

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

01 **CRD apiVersion discovery tool** NEW 55

A new `api_resources_list` tool lets users discover available CRD apiVersions across a cluster.

— Names the exact tool but no usage details or output format. v0.1.2

WAS THIS USEFUL? 👍 👎

pullmd

Feedback ⓘ v3.18.0 NOTES ↗ RANK

Self-hosted URL- and file-to-Markdown service for humans and AI agents - web pages, documents, images, audio, YouTube. PWA + REST + MCP + Claude Code skill, Reddit-aware, refreshable share links.

> pullmd's v3.18.0 release massively expands native file-upload format support (email, Office/OpenDocument, calendar, subtitles and more), moves CSV/JSON/XML/RSS parsing in-house away from the markitdown sidecar (a breaking change), and adds unlimited Reddit thread depth retrieval.

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

01 Expanded file upload format support in /api/file NEW

90

Adds native `POST /api/file` upload support for `.eml` emails (parsed via `postal-mime`, using the subject as title, running the HTML body through the same extraction as `POST /api/html`, with `frontmatter=true` filling `title`, `author`, `published` and `source: email`, `lang=en` switching header labels to English, and files without mail headers returning `422`), Outlook `.msg` files (via `@kenjiuno/msgreader`), Chrome/Edge saved pages `.mhtml` / `.mht` (exposing the stored URL as frontmatter `url` so site recipes apply), `.txt` / `.md` files (keeping existing YAML frontmatter as a fenced `yaml` block and decoding non-UTF-8 files as Windows-1252), subtitle files `.srt` / `.vtt` (converted to readable paragraphs with numbers, timestamps and styling removed, and rolling auto-caption repeats stripped), OpenDocument files `.odt` / `.ods` / `.odp` (headings, bold, italic, links, lists, tables, footnotes, one table per sheet, one section per slide with speaker notes), `.rtf` files (paragraphs, headings, bold, italic, links, lists and tables), calendar `.ics` files (start, end, location, organizer, attendees, recurrence and description per event, with invitations and cancellations marked), contact `.vcf` files (including vCard 2.1), Jupyter `.ipynb` notebooks, source/config/log files like `.py`, `.js`, `.yaml`, `.log` and Dockerfile (as a code block in the right language, deliberately excluding `.env`), and prose markup files `.rst`, `.adoc` and `.org` (handled like `.txt`).

Convert a saved e-mail to Markdown with structured frontmatter, ready for ingestion into a knowledge base.

```
$ curl -X POST https://your-pullmd-instance/api/file \
  -F 'file=@investigation-alert.eml' \
  -F 'frontmatter=true' \
  -F 'lang=en'
```

— Names every format and handling rule, plus a runnable upload example. v3.18.0

02 Unlimited Reddit thread depth via comment_depth=all NEW

90

Adds `comment_depth=all` support for retrieving full Reddit threads across `/api`, `/api/stream`, and the MCP `read_url` tool (`comment_depth: "all"`), loading load more comments and continue this thread placeholders back into place and lifting the Hacker News depth limit. A new `PULLMD_REDDIT_MORE_REQUESTS` environment variable controls how many extra Reddit requests are made when loading collapsed comments (set to `0` to disable), with any remaining missing comments noted below the thread.

Fetch a full Reddit threat-intel discussion thread with all collapsed comments expanded, using authenticated API credentials.

```
$ curl 'https://your-pullmd-instance/api?
url=https://www.reddit.com/r/netsec/comments/example/&comment_dep
th=all'
```

Cap the extra Reddit requests made to load collapsed comments to avoid rate limits in high-volume deployments.

```ini
PULLMD_REDDIT_MORE_REQUESTS=5
```

— Names endpoints, MCP tool, env var and includes runnable examples. v3.18.0

03 Native data file parsing replaces markitdown sidecar BREAKING

65

Uploaded `.csv` and `.tsv` files (delimiter auto-detected as a table), `.json` and `.xml` files (as a code block), and RSS/Atom feeds (as a list of entries) sent to `POST /api/file` are now read natively by pullmd instead of being handed to the markitdown sidecar. This is a breaking change: markitdown previously produced a table for CSV, so output format may differ after upgrading.

— Names the endpoint and formats but no migration example given. v3.18.0

└──▷ BREAKING ON UPGRADE

! Uploaded `.csv`, `.json` and `.xml` files are now read by pullmd itself instead of the markitdown sidecar; the markitdown sidecar previously produced a table for CSV, so output format may differ.

WAS THIS USEFUL? 👍 👎

mcp-server

Feedback ⓘ v2.6.0 NOTES ↗ RANK

BrowserStack's Official MCP Server

> mcp-server adds the ability to upload generic attachments from local environments to test management cases.

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

01 **Local attachment upload for test management** NEW 35

Adds a local-only capability to upload generic attachments from local environments directly to test management cases.

— Names the capability but no flag, command or endpoint given v2.6.0

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Perplexity API

Feedback ⓘ updated · Nov 5, 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 Decisions API now supports live web grounding and URL fetching, letting up to 100 questions per call be answered using fresh search results or freshly crawled pages.

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

01 Web grounding and URL fetching in Decisions API NEW

95

Setting `mode` to `"web_grounding"` on `POST /v1/decisions` runs a Fast Search for each question (up to 100) before the model answers, with results returned in `search_results`; each search costs $0.001 plus input tokens. Passing a page URL as `state` alongside `mode: "web_grounding"` fetches and inlines that page's text before search, returning the fetched page in `fetched_results` and search evidence in `search_results`, at an additional $0.001 per successful fetch. The `max_staleness` parameter sets a maximum page age in seconds — a stored copy older than the threshold triggers a fresh crawl.

Ground up to 100 decision questions against live web search results in a single API call.

```
$ curl -X POST https://api.perplexity.ai/v1/decisions \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"questions": ["Is X still the market leader?", "What is
the current price of Y?"], "mode": "web_grounding"}'
```

Fetch and inline a live page before grounding — and enforce a freshness window so stale cached copies trigger a re-crawl.

```
$ curl -X POST https://api.perplexity.ai/v1/decisions \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"questions": ["What does this policy say about data
retention?"], "mode": "web_grounding", "state": {"url":
"https://example.com/privacy-policy"}, "max_staleness": 86400}'
```

— Endpoint, params and costs all named with runnable examples. snapshot-20261105

WAS THIS USEFUL? 👍 👎

Hugging Face OpenEnv

Feedback ⓘ v0.12.0 NOTES ↗ RANK

An interface library for RL post training with environments.

> OpenEnv's v0.12.0 release adds a typed exception for dropped WebSocket sessions, structured JSON parsing for TextArena keyword environment variables, a required `--env-url` flag for the Wordle GRPO script, and a new Kubernetes deployment example.

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

01 JSON parsing for TEXTARENA_KW_* variables IMPROVED

65

`textarena_env` now parses `TEXTARENA_KW_*` environment variables as JSON, enabling structured keyword configuration instead of plain string values.

— Names exact env var pattern and module, usable directly. v0.12.0

02 Wordle GRPO script requires --env-url BREAKING

65

The Wordle GRPO script now requires the `--env-url` flag, making the target environment explicit; existing invocations that omit it will fail.

— Names exact flag and breaking impact clearly. v0.12.0

03 SessionLostError on WebSocket disconnect NEW

60

OpenEnv now raises `SessionLostError` when the WebSocket connection drops mid-episode, giving callers a typed exception to catch and handle connection loss instead of an unhandled failure.

— Names the exception but not catch/retry mechanics. v0.12.0

THINNER COVERAGE BELOW

04 Kubernetes echo deployment example NEW

30

Adds a Kubernetes echo example demonstrating how to deploy environments on Kubernetes.

— Only a bare mention, no path or commands given. v0.12.0

└──▷ **BREAKING ON UPGRADE**

! The Wordle GRPO script now requires `--env-url`; existing invocations that omit it will fail.

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK Feedback ⓘ v7.51.0 NOTES ↗

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

OpenAI TypeScript SDK adds GPT-6 Rosalind to the model catalog, GIF and SVG image output with complexity controls, and expanded agent turn errors.

└──▷ GET THIS VERSION ▾

npm

```
$ npm install openai@7.51.0
```

git

```
$ git clone --branch v7.51.0 https://github.com/openai/openai-node.git
# already have the repo? check out this version:
$ git checkout v7.51.0
```

signed build provenance · checked 5 Nov Wrong command?

Generate images as GIF or SVG output formats and control image complexity via new API fields.

Use GPT-6 Rosalind as a model identifier in the model catalog.

Handle expanded agent turn error types and additional image and audio options in API responses.

NEW SUPPORTED PLATFORM Australia, India, and Japan data residency options

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.46.0 NOTES ↗

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

OpenAI Python SDK adds GPT-6 Rosalind to the model catalog, GIF/SVG image output formats, image complexity controls, and expanded agent turn error types.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.46.0 https://github.com/openai/openai-python.git
# already have the repo? check out this version:
$ git checkout v3.46.0
```

Add `gpt-6-rosalind` to the model catalog for use in API calls.

Add image complexity controls and GIF and SVG output format support to image generation.

Expand agent turn error types and broaden image and audio options in the API.

NEW SUPPORTED PLATFORM Australia, India, and Japan data residency options

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v2.5.0 NOTES ↗ RANK

The Anthropic Python SDK provides typed Python clients for Claude APIs, streaming responses, and tool use.

> The Anthropic Python SDK adds structured, precise error handling and a new workspace filtering option.

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

01 Structured error classes with detailed fields NEW

80

New error classes keyed to HTTP status codes and API error codes let callers catch HTTP and API errors precisely. `APIStatusError` now exposes `error_code`, `details`, and `api_message` fields, and new error details types let callers inspect structured failure reasons.

Distinguish rate-limit or auth failures from server errors in production error-handling code.

python

```python
from anthropic import Anthropic, APIStatusError

client = Anthropic()

try:
    msg = client.messages.create(
        model="claude-opus-5-5",
        max_tokens=1024,
        messages=[{"role": "user", "content": "Hello"}],
    )
except APIStatusError as e:
    print(f"HTTP status : {e.status_code}")
    print(f"API error code : {e.error_code}")
    print(f"API message    : {e.api_message}")
    print(f"Details        : {e.details}")
```

— Names fields and shows runnable try/except example v2.5.0

THINNER COVERAGE BELOW

02 **ID filter for listing workspaces** NEW 40

The list workspaces API gains a new `ids` parameter to filter workspace list results by ID.

— Names parameter but no usage example or endpoint path v2.5.0

WAS THIS USEFUL?  

Anthropic TypeScript SDK

Feedback ⓘ sdk-v1.5.0

NOTES ↗ RANK

The Anthropic TypeScript SDK provides typed JavaScript and TypeScript clients for Claude APIs, streaming, and tool use.

> The Anthropic TypeScript SDK v1.5.0 adds typed error classes with structured failure details and a new filter for listing workspaces.

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

01 Typed error classes and structured error details NEW

72

New error classes are thrown for specific HTTP statuses and API error codes, letting callers catch errors precisely. `APIError` now exposes `apiMessage`, `errorCode`, and `details`, and new error details types cover specific error categories with named types for inspecting structured failure context.

Handle specific API error conditions by catching typed error classes and reading structured details from `APIError`.

typescript

```typescript
import Anthropic, { APIError } from '@anthropic-ai/sdk';

const client = new Anthropic();

try {
  const message = await client.messages.create({
    model: 'claude-opus-5-5',
    max_tokens: 1024,
    messages: [{ role: 'user', content: 'Hello' }],
  });
} catch (err) {
  if (err instanceof APIError) {
    console.log('HTTP status:', err.status);
    console.log('Error code:', err.errorCode);
    console.log('API message:', err.apiMessage);
    console.log('Details:', err.details);
  }
}
```

— Names exact fields and includes runnable catch example sdk-v1.5.0

THINNER COVERAGE BELOW

02 **Filter workspace list by ID** NEW 35

The list workspaces API now accepts an `ids` parameter to filter results by workspace ID.

— Names the parameter but no usage example sdk-v1.5.0

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore Feedback ⓘ

Runtime: Platform version V2 now available in five additional Regions (2026-11-04)

BLOG ↗

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

NEW SUPPORTED PLATFORM Amazon Bedrock AgentCore Runtime platform version V2 — additional region availability

WAS THIS USEFUL? 👍 👎

MotleyAI SLayer

Feedback ⓘ v1.6.0 NOTES ↗ RANK

SLayer is an embeddable semantic layer that lets AI agents and people query warehouse data through shared metric definitions.

> SLayer v1.6.0 adds a `client_factory` parameter for routing database access through a custom gateway across its embedding surfaces, plus a synchronous ingestion method for simpler scripting.

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

01 Custom client_factory for database access routing NEW

78

Pass a `client_factory` argument to `SlayerQueryEngine`, `SlayerClient`, `create_app`, `create_mcp_server`, the Postgres facade, or the Flight server to route all database access through your own gateway, keeping database secrets out of application code.

Embed SLayer behind a credential gateway so database secrets never touch your application code directly.

python

```
from slayer.engine.query_engine import SlayerQueryEngine

engine = SlayerQueryEngine(client_factory=my_gateway_factory)
result = engine.query({"source_model": "orders", "measures":
["sum(order_total)"]})
```

— Names six integration points and a runnable code example v1.6.0

02 Synchronous ingestion method engine.ingest_sync() NEW

68

Ingest data from Python in one line using engine.ingest_sync(ds, schema="public") for synchronous contexts, or await engine.ingest(ds, ...) for async, on the query engine.

Trigger a one-shot ingestion of a datasource from a Python script or task runner without an async event loop.

python

```
engine.ingest_sync(ds, schema="public")
```

— Exact method signature and runnable one-liner given v1.6.0

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

Eigen Labs Darkbloom

Feedback ⓘ v0.10.8 NOTES ↗

CODE ↗ RANK

Darkbloom is a for the innovators pushing the frontier.

> Darkbloom's v0.10.8 release adds a full Mac uninstall command, extends cluster commands with pairing-aware behavior and a `--solo` flag (with a breaking change to plain `cluster stop`), and adds scriptable JSON output to cluster status plus a rollback option on update.

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

01 darkbloom uninstall command NEW

80

A new `darkbloom uninstall` command removes Darkbloom from a Mac, with `--everything` or `--keep-models` to control whether downloaded models are deleted, `--yes` to skip confirmation, and `--dry-run` for scripted previews.

Remove Darkbloom from a Mac completely in a scripted, non-interactive way (for fleet offboarding).

```
$ darkbloom uninstall --everything --yes
```

Remove Darkbloom but keep downloaded models so a reinstall skips re-downloading them.

```
$ darkbloom uninstall --keep-models --yes
```

— Exact flags and runnable commands given for each outcome. v0.10.8

02 Pair-aware cluster start/stop/restart/status BREAKING

80

Cluster `start`, `stop`, `restart`, and `status` commands are now aware of paired Macs, with a new `--solo` option to target only the local machine. As a breaking change, plain `darkbloom cluster stop` now stops serving on both Macs instead of only the local Mac's cluster service — anywhere prior guidance said `darkbloom cluster stop` for a rollback or cluster-service restart, use `darkbloom cluster stop --service` instead.

— Clear before/after and exact migration flag, but no --solo example. v0.10.8

03 **`darkbloom cluster status --json`** NEW 70

`darkbloom cluster status --json` returns machine-readable cluster status for scripts and agents and sets an exit code; the previous verbose report is still available via `--details`.

Poll pair cluster status from a script or agent and branch on the exit code.

```
$ darkbloom cluster status --json
```

— Named flags and a runnable example, but exit-code semantics unspecified. v0.10.8

THINNER COVERAGE BELOW

04 **`--rollback` flag on update** NEW 20

The `update` command gains a `--rollback` option, per the release summary; no further behavior is described.

— Only named in the release summary, no detail or example given. v0.10.8

└──▷ **BREAKING ON UPGRADE**

! Plain `darkbloom cluster stop` now stops serving on both Macs instead of only the local Mac's cluster service — anywhere v0.10.7 guidance says `darkbloom cluster stop` for a rollback or cluster-service restart, use `darkbloom cluster stop --service` instead.

WAS THIS USEFUL? 👍 👎

Fireworks AI

Feedback ⓘ Cookbook migration and first PyPI release

NOTES ↗ RANK

Fireworks AI provides hosted inference, model fine-tuning, and deployment APIs for open-weight models.

> Fireworks AI packaged its training cookbook for direct PyPI installation and consolidated the public cookbook into a new repository.

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

01 Training cookbook installable via PyPI with repo migration

NEW 83

Training recipes and utilities can now be installed directly with `python -m pip install --upgrade fireworks-training-cookbook` (version `2026.11.5.1`, requires Python 3.11+), with imports remaining under the `training.*` namespace. The public cookbook itself has moved to the `fw-ai-external/cookbook` repository, which now hosts recipes, case studies, notebooks, and coding-agent skills, and agent skill install commands have been updated to point to this new repository.

— Exact pip command, version, Python requirement, and repo path given.

Cookbook migration and first PyPI relea…

WAS THIS USEFUL? 👍 👎

Lambda Cloud

Feedback ⓘ announced · Nov 4, 2026 BLOG ↗ RANK

Lambda provides on-demand GPU cloud instances and clusters for AI training and inference.

> Lambda Cloud launched a full identity and access management system in the console, adding role-based access control, service identities, groups, and audit logging in a single release.

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

01 **Service identities for scripts and pipelines** NEW 85

Service identities are accounts for scripts and automation, created under the Service identities tab in `Account > Configuration > IAM`; each gets its own API keys, roles, and an optional key expiration date, and can hold up to two keys simultaneously for zero-downtime rotation. Lambda Cloud now supports up to 50 service identities per account.

Replace a personal API key used by an automation script with a least-privilege service identity so the script can no longer exceed its intended access.

📍 1. Open Account › Configuration › IAM in the Lambda Cloud console. 2. Go to the Service identities tab and create a new service identity for your script. 3. On the identity's details page, assign only the roles the script requires (e.g. Instance Reader or Workspace Member). 4. Generate an API key for the identity and set an expiration date. 5. Update the script to use the new key, then delete the old personal API key from your user page.

— Concrete limits and rotation mechanism with a runnable migration path

`launch-20261104-e93f9f19`

02 **RBAC roles and effective access view** NEW 77

The Lambda Cloud console gains an IAM system accessed via `Account > Configuration > IAM`, with predefined roles — Account Admin, IAM Admin, Billing Admin, Workspace Owner, Workspace Member, and Instance Reader — assignable at account or workspace scope to users, service identities, or groups. An effective access view on an identity's details page lets you inspect every role it holds across workspaces, including roles inherited from group membership.

Audit an identity's actual permissions before granting it access to a sensitive workspace, to catch quietly accumulated access from group memberships.

📍 1. Open Account › Configuration › IAM and navigate to the user or service identity. 2. Open the identity's details page and select the effective access view. 3.

Review every role the identity holds in each workspace, including any inherited from groups.

— Names roles and scope, gives exact console path and audit steps

launch-20261104-e93f9f19

03 **Groups for role and workspace inheritance** NEW 72

Users and service identities can be organized into groups (up to 100 per account) via the Groups tab; a role granted to a group is inherited by all its members, and workspace membership can be managed at the group level.

Simplify onboarding and offboarding by grouping team members so a single role assignment covers everyone on the team.

📍 1. Open Account › Configuration › IAM and go to the Groups tab. 2. Create a group for your team (e.g. ml-infra-team). 3. Assign the appropriate roles (e.g. Workspace Member) to the group. 4. Add the group to the relevant workspace. 5. Add new hires to the group to grant access, or remove departing members from the group to revoke it.

— States limit and inheritance mechanism with onboarding steps launch-20261104-e93f9f19

THINNER COVERAGE BELOW

04 **Audit logs for IAM activity** NEW 35

Audit logs covering keys, identities, workspaces, and resources are recorded automatically and retained for six months.

— Names scope and retention but no access path or usage detail launch-20261104-e93f9f19

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

Mozilla AI Otari

Feedback ⓘ v0.38.0 NOTES ↗ RANK

Open-source, OpenAI-compatible LLM gateway you run yourself. One endpoint for 40+ providers, with virtual keys, budgets, and usage tracking.

> Otari v0.38.0 adds an image edit endpoint and trace retention for routed request content, while tightening search/fetch billing enforcement and aligning price unit naming with models.dev conventions.

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

01 Search billing treats read pages as fetches IMPROVED

45

Search tool usage now bills the pages a search reads as fetches, charged on the instance that read them.

– Mechanism described but no pricing config or command shown. v0.38.0

02 Refusal of searches reading unpriced fetch pages BREAKING

45

Otari now refuses a search whose pages an unpriced fetch tool would read. This is a breaking change: any workflow relying on such searches passing through will now fail.

– Breaking behavior explained but no migration or override path given. v0.38.0

03 Price unit naming aligned with models.dev BREAKING

45

Price units in the dashboard now match the API naming convention, aligned with models.dev conventions; this is a breaking change for any configuration or integration relying on the previous price unit names.

– Names the convention but not the specific old/new unit strings. v0.38.0

04 Trace retention for routed request content IMPROVED

35

Otari now keeps the content of routed requests that ask for it in traces.

– No flag or config key for enabling this retention. v0.38.0

05 **Image edit endpoint added** NEW

25

Otari adds an image edit endpoint, introduced in PR #2868.

— No endpoint path, method, or usage details given. v0.38.0

└──▷ **BREAKING ON UPGRADE**

! Price unit spellings are now aligned with models.dev conventions — any existing configuration or integration that relied on the previous price unit names will break.

! Searches whose pages an unpriced fetch tool would read are now refused — any workflow relying on those searches passing through will break.

WAS THIS USEFUL?  

dario

Feedback ⓘ v6.21.0 NOTES ↗ RANK

One local endpoint that routes Cursor, Cline, Aider, Claude Code and the Agent SDK through your Claude and ChatGPT subscriptions, with failover and pooling and zero runtime dependencies.

> dario introduces fleet-wide shared pending-login state for multi-replica deployments and fixes a Codex login race that could silently overwrite an in-use seat.

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

01 Cross-replica admin login completion via --pool-shared-accounts

NEW 91

The new `--pool-shared-accounts` flag stores pending `/admin/login/start` and `/admin/codex/login/start` state on the refresh-lock service (Redis) so a login started on one replica behind a load balancer can be completed on any other, avoiding 410 errors. State is single-use, expires after 10 minutes, and is capped fleet-wide at 64 pending logins per engine; enabling it requires the refresh-lock service to serve new `/logins/*` endpoints, meaning a Redis image rebuild or Worker redeploy before turning the flag on — an older Worker without these endpoints falls back to per-instance storage and logs a warning.

Run dario behind a load balancer with multiple replicas so that an admin login started on replica A can be completed on replica B, without 410 errors.

```
$ dario --pool-shared-accounts
```

— Mechanism, limits and runnable command all given v6.21.0

THINNER COVERAGE BELOW

02 Codex login conflict returns 409 instead of overwriting seat

IMPROVED 48

A Codex login now returns `409` instead of silently overwriting a seat when the completing instance already holds one under the alias; the login is left unconsumed so a different instance can complete it instead. This applies regardless of whether `--pool-shared-accounts` is enabled.

— Clear mechanism and status code, no usage example v6.21.0

└──▷ **BREAKING ON UPGRADE**

! When `--pool-shared-accounts` is enabled, the new `/logins/*` endpoints must be available on the refresh-lock service (Redis Worker); an older Worker that does not serve them causes logins to fall back to per-instance storage and logs a warning. Rebuild the Redis image or redeploy the Worker before enabling the flag.

WAS THIS USEFUL?  

◆ Local LLM Runtimes

llama.cpp

Feedback ⓘ v0.9.0 NOTES ↗ CODE ↗ RANK

Llama CPP is a C/C++ library and set of command-line tools that runs large language models locally.

> This release adds OCP FP8 E4M3 quantization, full Chatterbox TTS support with predefined voices, two new model architectures, and multi-node tensor parallelism via NCCL, alongside a batched tensor copy API in ggml and improvements to the server's `/v1/responses` and `/props` endpoints.

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

01 **Chatterbox TTS with predefined voices** NEW 90

Adds Chatterbox TTS support throughout the mtmd audio pipeline: `MTMD_GEN_AUDIO_TYPE_CHATTERBOX` and `n_voices` in `mtmd_gen_audio_info` (`mtmd.h`), plus new APIs mtmd_gen_audio_voice_name(), mtmd_gen_audio_get_voice(), and mtmd_gen_state_set_ref() for selecting predefined voices without voice cloning, and mtmd_bitmap_is_voice() and mtmd_gen_audio_read_tensor() for audio bitmap and tensor handling. The server's `GET /props` endpoint now reports a model's predefined TTS `voices`, and the TTS endpoint accepts predefined voice names directly.

Query the server for a model's available predefined TTS voices before submitting a synthesis request.

```
$ curl http://localhost:8080/props
```

— Full named API surface with a runnable query example v0.9.0

02 **extra_params merging in /v1/responses** IMPROVED 81

The server `/v1/responses` endpoint now merges `extra_params` (such as `seed`, `min_p`, and others) into the converted chat-completions response, letting callers pass extra sampling parameters through the OpenAI-compatible endpoint.

Pass extra sampling parameters such as seed and min_p through the OpenAI-compatible responses endpoint.

```
$ curl http://localhost:8080/v1/responses -H 'Content-Type:
application/json' -d '{"model":"my-
model","input":"Hello","extra_params":{"seed":42,"min_p":0.05}}'
```

— Named endpoint and fields with a runnable curl example v0.9.0

03 **OCP FP8 E4M3 quantization support** NEW 73

Adds `LLAMA_FTYPE_MOSTLY_F8_E4M3` to `llama.h`, enabling OCP FP8 E4M3 (saturating mode) quantization with conversion helpers and a new ftype.

— Named flag and format given, no usage example v0.9.0

04 **Batched tensor copy API in ggml** NEW 73

Adds a batched tensor copy API in ggml — `ggml_backend_tensor_copy_n`, `ggml_backend_tensor_copy_set_n`, and `ggml_backend_tensor_copy_get_n` — superseding the previous `_2d` API, implemented for the Metal and CUDA backends.

— Named functions and backends but no usage example v0.9.0

THINNER COVERAGE BELOW

05 **Qwen3-VL video prompt template support** NEW 45

Adds `MTMD_VIDEO_TEMPLATE_QWEN3VL` to `mtmd.h` for accurate Qwen3-VL video prompt format support.

— Named constant but no usage detail given v0.9.0

06 **New model architecture support** NEW 40

Adds support for the Clover-35B-A3B-Preview diffusion model running in AR mode, and for the Limite 1B - Violetto model architecture.

— Model names given but no integration or usage detail v0.9.0

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

vLLM

Feedback ⓘ announced · Nov 4, 2026 BLOG ↗

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

vLLM adds Kimi K3.1 Pro support with DSA sparse attention, token-sharded prefill, DCP with exact top-k merge, mixed KV quantization, and zero-mismatch RL.

NEW SUPPORTED PLATFORM Serve Kimi K3.1 Pro with `vllm serve moonshotai/Kimi-K3.1-Pro` using `--tool-call-parser kimi_k3`, `--reasoning-parser kimi_k3`, and `--load-format fastsafetensors` on NVIDIA GPUs, AMD GPUs, Google TPUs, and Alibaba T-Head Zhenwu PPUs.

Enable speculative decoding for Kimi K3.1 Pro via `--speculative-config` with `'method':'dspark'` and `'draft_sample_method':'probabilistic'` using the `Inferact/Kimi-K3.1-Pro-DSpark-preview` draft model.

vLLM now supports Kimi K3.1 Pro's Dedicated Sparse Attention (DSA) layers, packing MLA latent, indexer keys, and KDA state into a single contiguous block for use with FlashInfer and FlashMLA sparse kernels.

Token-sharded prefill attention now redistributes query tokens across GPUs via all-to-all instead of splitting by head, reaching up to 1.69× faster time to first token at 340K tokens on 8× GB300.

Decode context parallelism (DCP) with DSA now uses an exact top-k merge that sends ~10× fewer bytes, making the attention module 1.3–1.5× faster at DCP8 versus a generic merge.

+4 more

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

uteke

Feedback ⓘ v0.23.0 NOTES ↗ RANK

The Brain for Your AI — Local-first memory engine for AI agents. Store, recall, and search memories with semantic embeddings.

> uteke v0.23.0 ships temporal (time-travel) recall with valid_from/valid_until support, trust-tier and similarity filtering on recall, a read-only browser viewer with a new Sessions tab, namespace migration tooling, Markdown export with front-matter, richer Cloud-import reporting, and agent hook path pinning — alongside breaking changes to `--at` defaults, `--strict` scoring, and MCP output fencing.

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

01 Temporal validity and time-travel recall

BREAKING 95

uteke adds temporal validity to memories: `uteke recall --at T --history` and `uteke list --at T --history` filter results to memories valid at a past instant, including those deprecated after that point. Callers can store `valid_from` and `valid_until` (RFC 3339) via `POST /remember`, `PUT /memory`, and the MCP tools `uteke_remember` and `uteke_update`. Breaking: `recall --at T` and `list --at T` now default to history mode, whereas previously deprecated memories were excluded from `--at` results.

Recall what the agent believed at a past point in time, including facts that were later superseded or forgotten — useful for auditing what context an agent had during an incident.

```
$ uteke recall --at 2025-11-03T09:00:00Z --history 'deployment configuration'
```

— Runnable commands and endpoints with explicit before/after behavior. v0.23.0

02 Trust-tier and similarity filtering on recall

BREAKING 95

Recall results gain `trust_tier` and `similarity` fields: filter by provenance tier with `uteke recall --min-trust <tier>` (values `human`, `agent`, `external`, or `untrusted`) or the `min_trust` field on `POST /recall` and `uteke_recall`; inspect vector-leg cosine similarity (0 to 1, or `null` for full-text-only hits) via the `similarity` field on the CLI with `--json`, `POST /recall`, and `uteke_recall`. `recall --strict` and `[recall] min_score_strict` now drop hits with no vector

match by comparing against `similarity` (unboosted cosine) instead of `score`, so callers relying on the old `score`-based comparison will see different, now non-empty, result sets.

Return only recall results backed by a vector match, filtered to human-sourced memories, to surface high-confidence human-authored context for a sensitive decision.

```
$ uteke recall --strict --min-trust human 'access control policy'
```

— Multiple named fields, flags, and endpoints with explicit behavior change. v0.23.0

---

03 **Read-only loopback viewer via uteke-serve** NEW 85

`uteke-serve --viewer` launches a read-only loopback viewer for browsing memories, namespaces, rooms, and health. Every request requires a per-launch token and a loopback `Host`, only `GET` is served, and a non-loopback bind requires the server auth token.

— Names flag, HTTP method, and auth mechanism in detail. v0.23.0

---

04 **Namespace migration via TOML map** NEW 83

`uteke namespace migrate --map <file>` moves whole namespaces into `default` using a TOML map file; it dry-runs by default, writes changes with `--apply`, and soft-deletes namespaces marked `drop = true` in the map (requires `--allow-drop`).

— Runnable command with all flags and safety defaults named. v0.23.0

---

05 **Uteke Cloud import reporting** IMPROVED 80

`uteke import` and `POST /import` import a Uteke Cloud export including documents, rooms, and memories; the result now reports `documents_imported`, `documents_updated`, `documents_unchanged`, `parent_unresolved`, `rooms_imported`, `rooms_existing`, `room_documents_linked`, and `room_documents_missing`.

— Command and endpoint named with full list of result fields. v0.23.0

---

06 **Markdown export with YAML front-matter** NEW 75

`uteke doc export --front-matter` exports Markdown documents with YAML front-matter (`slug`, `title`, `tags`, `parent`); the plain export is unchanged and `--front-matter` is ignored when combined with `--json`.

— Exact flag and front-matter fields named with interaction caveat. v0.23.0

---

07 **Absolute path pinning for agent hooks** NEW 70

`uteke connect <agent> --absolute-path` pins the running binary path into hook commands so agents do not pick up an earlier `PATH` entry named `uteke`.

Wire Claude Code into uteke so every session starts with recalled context and each prompt and reply is captured automatically.

```
$ uteke connect claude
```

— Exact flag and command given; rationale stated but no deeper mechanism. v0.23.0

---

08 **Sessions tab in uteke view** NEW 70

A new Sessions tab on the `uteke view` page shows captured sessions and their turns in `seq` order, with each memory whose source is `capture:<session>:<batch>` or `observation:<session>` linking back to its session.

— Clear UI path but no CLI/API surface to act on directly. v0.23.0

---

09 **MCP output fencing toggle** BREAKING 70

MCP read tools now wrap returned text in `<uteke-stored-data>` by default; set the environment variable `UTEKE_MCP_FENCE=0` to disable this fencing and restore the previous plain-text output for clients that parse the text.

— Named env var and exact behavior change, minimal extra mechanism. v0.23.0

└──▷ BREAKING ON UPGRADE

! `recall --at T` and `list --at T` now default to history mode (also returning memories deprecated after T); previously deprecated memories were excluded.

! `recall --strict` and `[recall] min_score_strict` now compare against the `similarity` field (unboosted cosine) instead of `score`; callers that relied on the old `score`-based comparison will see different — and now non-empty — result sets.

! MCP read tools now wrap returned text in `<uteke-stored-data>` by default; set `UTEKE_MCP_FENCE=0` to restore the previous plain-text output for clients that parse the text.

WAS THIS USEFUL? 👍 👎

Weaviate

Feedback ⓘ v1.39.15 NOTES ↗ RANK

Weaviate is an open-source vector database that stores both objects and vectors, allowing for the combination of vector search with structured filtering with the fault tolerance and scalability of a cloud-native database.

> Weaviate's hybrid search REST endpoint now lets callers supply their own vector directly, bypassing server-side vectorization.

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

01 **Caller-supplied vector for hybrid search** IMPROVED 50

The hybrid search REST endpoint now accepts a caller-supplied vector directly, letting callers bypass server-side vectorization for hybrid queries.

— Names the endpoint and mechanism but no request details or examples v1.39.15

WAS THIS USEFUL? 👍 👎

Crow Desktop

Feedback ⓘ v0.1.5 NOTES ↗ RANK

Crow: build your second brain from the documents on your computer. Downloads for macOS, Windows and Linux.

> Crow Desktop's biggest change is AI-driven folder management powered by llmwiki2 0.8.0, shipped alongside several wiki-filing improvements: a 10-folder cap with overflow handling, skip-on-edit protection, automatic external link conversion, and faster filing at scale.

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

01 **AI-driven folder proposals via llmwiki2 0.8.0** NEW

55

AI apps can now open a named top-level folder, propose a new folder with a preview of the notes it would hold and create it on confirmation, or re-file every note from Crow's stored copies on confirmation. This is powered by llmwiki2 0.8.0.

— Named mechanism and version but no API/command shown v0.1.5

02 **10-folder cap with overflow filing** NEW

45

Crow Desktop now caps wikis at 10 top-level folders; documents that would create an eleventh are filed in the closest existing folder instead.

— Clear mechanism and exact limit, but automatic not user-driven v0.1.5

03 **Skip-on-edit protection for active notes** NEW

42

Crow Desktop now skips merging into a note you are actively editing and re-files it at the next index run, so in-progress edits are never overwritten.

— Explains behaviour and timing but no user-facing control v0.1.5

04 **Automatic plain-text conversion for external links** IMPROVED

26

Links in a note to files outside the wiki are now converted to plain text automatically.

— Simple statement, no mechanism or scope detail v0.1.5

05 **Faster filing performance at scale** IMPROVED

13

Filing performance now degrades far less as the wiki grows.

— No numbers or benchmarks given, vague improvement v0.1.5

WAS THIS USEFUL? 👍 👎

infino

Feedback ⓘ v0.11.23 NOTES ↗ RANK

Embedded retrieval library built on Parquet. Fast, efficient, and scalable.

> Infino's vector_search now supports filtering while maintaining performance even when the filter matches a large number of documents.

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

01 **Filtered vector search with sustained performance** IMPROVED

50

`vector_search` now supports filtering and maintains performance when the filter keeps a large number of matches.

– Names the API but no mechanism or benchmark details v0.11.23

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.147.0 NOTES ↗ RANK

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

> Docling shipped a single consistency fix ensuring inline text spacing follows the InlineGroup whitespace contract across all backends.

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

01 **Consistent InlineGroup whitespace handling across backends**

IMPROVED

35

Docling now honors the `InlineGroup` whitespace contract across all backends, producing consistent whitespace handling for inline content.

– Names the contract but no mechanism or scope detail v2.147.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

Arize Phoenix

Feedback ⓘ arize-phoenix-client-v3.7.0 NOTES ↗

RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix now captures model parameters and agent settings as part of experiment metadata for Harbor experiments.

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

01 **Model parameters and agent settings in experiment metadata**

NEW 30

Arize Phoenix now records model parameters and agent settings in experiment metadata for Harbor experiments.

— Names the capability but no mechanism, UI path, or API detail.

arize-phoenix-client-v3.7.0

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.140 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik expanded its catalog settings panel to support two additional LLM providers.

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

01 **Gemini and Vertex AI provider support** NEW 40

The catalog settings panel now supports Gemini and Vertex AI, letting users configure these providers alongside existing ones.

— Names panel and providers but no config details 2.2.140

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.75.0 NOTES ↗ RANK

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

> Langfuse's v4.75.0 release streamlines Codex integration with the AI Gateway, gives gateway-only members their own usage visibility, and substantially raises AI Gateway rate limits for paid plans.

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

01 **Codex setup simplification and skill tracing** IMPROVED

55

Codex setup is simplified to an exported key plus a `config.toml` for AI Gateway configuration, and Langfuse now traces Codex skill resource loads for observability into Codex skill execution.

— Names config.toml and key export but lacks further mechanism detail v4.75.0

02 **AI Gateway member usage view and tile** NEW

35

Gateway members can now view analytics of their own usage within the AI Gateway, and Langfuse shows a dedicated AI Gateway tile on the organizations overview for gateway-only members.

— Describes what's visible but not how to access or configure it v4.75.0

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ updated · Nov 5, 2026 NOTES ↗

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

Braintrust adds parent-span trace continuation via `--parent` on `bt trace run`, optional MCP telemetry, and structured filters in the search dropdown.

Make `bt mcp run` calls without a telemetry argument — the argument is now optional, so you pass only the tool's own arguments.

Continue a coding-agent trace from an existing span by passing `--parent` to `bt trace run` with the value from an SDK span's export(), attaching the session under that span instead of starting a new trace.

The search dropdown's Recent section now includes structured filters alongside SQL filters, so you can reapply a recent filter without rebuilding it.

UPDATED INTEGRATION Anthropic Java SDK tracing — 3.x support

NEW INTEGRATION Temporal activity tracing

UPDATED INTEGRATION OpenAI Responses API span fields

UPDATED INTEGRATION Cursor SDK tracing — Context Summary spans

UPDATED INTEGRATION Gemini thinking-token span fields

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

ToolHive

Feedback ⓘ v0.62.0 NOTES ↗ RANK

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

> ToolHive's v0.62.0 release adds Codex for Bedrock model support, a TLS listener option for the Kubernetes operator, and two virtual MCP server updates that align it with the final MCP 2026-07-28 spec.

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

01 **Codex for Bedrock models in llm setup** NEW 50

The `llm setup` command can now configure Codex for Bedrock models, adding a new supported model provider to ToolHive's LLM setup workflow.

— Names the command and model but no further mechanism. v0.62.0

02 **URL elicitation for tools/call in vMCP** IMPROVED 50

The virtual MCP server now answers `tools/call` requests with a URL elicitation, aligning its behavior with the final MCP 2026-07-28 spec.

— Names the exact request type and spec version aligned to. v0.62.0

03 **TLS listener field for operator** NEW 40

The ToolHive operator configuration gains a new TLS listener field, along with validation for it, allowing a TLS listener to be set on the operator.

— Names the config field but not its exact key or usage. v0.62.0

WAS THIS USEFUL? 👍 👎

cplt

Feedback ⓘ 2026.11.04-120433-8bf5e16 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 default-on `brief` mode for new configs and a new `-b` flag for branch handling in sandboxed worktrees.

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

01 **Branch checkout flag for worktree subcommand** NEW 65

The `worktree` subcommand gains a `-b` flag to check out a local branch or create a new one inside a sandboxed worktree.

— Names exact flag and subcommand, runnable as-is 2026.11.04-112710-87fcde1

THINNER COVERAGE BELOW

02 **Brief mode enabled by default** NEW 40

cplt now enables `brief` mode by default for new configurations, with the setting loaded by Copilot and OpenCode agents.

— Names the setting and agents but not its effect 2026.11.04-120433-8bf5e16

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Google Gemini API

Feedback ⓘ updated · Nov 5, 2026 NOTES ↗

RANK

Gemini API provides Gemini models for generating text, images, and other multimodal content in applications.

> Gemini API expanded what tools can send back to the model (Office files, audio, video via function responses), added a voice remixing endpoint, and widened document format support to PPT, ODP, ODS, and DOC.

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

01 **Voice remixing endpoint for TTS voices** NEW 90

Voice remixing is now available via `POST /v1beta/voices` with `gemini-3.8-flash-tts`, letting you modify an existing prebuilt or designed voice using natural-language instructions to create a new stored voice with a `voice_...` ID.

— Exact endpoint, model, and ID format given snapshot-20261105

02 **Rich media returned via function responses** NEW 85

Tools can now return Office and OpenDocument files, audio, and video to the model via function responses, in addition to images, PDFs, and plain text. The `GenerateContent` API now accepts `fileData` in function response parts so tools can return a Files API URI instead of inline bytes; `displayName` is now available on both `inlineData` and `fileData`, and video parts accept `mediaProcessing` set to `STATIC` or `AGENTIC`. The Interactions API `function_result` steps now accept `document`, `audio`, and `video` content blocks in addition to `text` and `image`, by inline `data` or `uri`; video blocks in `v1beta` and `v1alpha` accept a `processing` field.

— Names exact fields and APIs but no runnable endpoint snapshot-20261105

03 **Expanded document format support (PPT, ODP, ODS, DOC)** NEW 65

Gemini API now accepts PowerPoint 97-2003 (PPT), OpenDocument Presentation (ODP), and OpenDocument Spreadsheet (ODS) files as input, and adds page-image processing for Word 97-2003 (DOC) files. ODS files provide the model with data and formulas from the first sheet, while DOC, PPT, and ODP files are rendered as page images.

— Clear format list and processing behaviour, no API detail snapshot-20261105

WAS THIS USEFUL? 👍 👎

ricursiveAI/Ripple-65B-Preview Feedback ⓘ

released · Nov 5, 2026 MODEL ↗

ricursiveAI/Ripple-65B-Preview from ricursiveai was released on 4 Nov 2026. 1049K-token context; $0.20 in / $0.60 out per million tokens (OpenRouter).

WAS THIS USEFUL? 👍 👎

Inception Mercury 2 Feedback ⓘ

retirement scheduled · Nov 4, 2026 MODEL ↗

Mercury 2 is a chat model from Inception.

OpenRouter lists a retirement date of 8 Nov 2026 for inception/mercury-2, first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

Inception Mercury 2.5 Feedback ⓘ

retirement scheduled · Nov 4, 2026 MODEL ↗

Mercury 2.5 is a chat model from Inception.

OpenRouter lists a retirement date of 8 Nov 2026 for inception/mercury-2.5, first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

Inception Mercury 3

Feedback

retirement scheduled · Nov 4, 2026

MODEL ↗

Mercury 3 is a chat model from Inception.

OpenRouter lists a retirement date of 8 Nov 2026 for inception/mercury-3, first seen 4 Nov 2026.

WAS THIS USEFUL?

Perceptron

Feedback ⓘ announced · Nov 4, 2026 BLOG ↗ RANK

Perceptron is a layer of intelligence for the physical world. We are a research company building the future of Physical AGI.

> Perceptron launched Egocentric 1.0, a self-service API that turns first-person human and robot video into structured robotics training data, backed by a new Mk1.5-based 3D hand reconstruction model.

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

01 **Egocentric 1.0 self-service video-to-training-data API** NEW 76

Perceptron launched Egocentric 1.0, a self-service API that accepts first-person human and robot video from household, robot arm, and manufacturing contexts and returns structured robotics training data: timestamped subtask timelines, continuous per-hand activity labels, 2D keypoints, and 3D hand mesh reconstruction. Pricing is $9 per video-hour at the default 4 fps, or approximately $15 per video-hour at 8 fps.

— Names outputs and pricing but no API endpoint or auth details launch-20261104-924d2e70

02 **3D hand reconstruction model built on Mk1.5** NEW 70

Perceptron added a new 3D hand reconstruction model, built on Mk1.5, that estimates each hand's shape, finger articulation, and camera-relative position without requiring camera calibration. It achieves 12–25% lower hand-shape error than ACE-Ego-Hand across all four benchmark settings on the HOT3D and ARCTIC datasets.

— Clear mechanism and benchmark numbers but no direct usage path given

launch-20261104-924d2e70

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

Cohere Models

Feedback ⓘ November 4, 2026 NOTES ↗

Cohere develops Command, Aya, Embed, Rerank, transcription, and document-understanding models.

Cohere adds Embed 5 and Rerank 5 models to OpenRouter; Embed 5 accepts text, images, and mixed text-and-image inputs including PDF pages.

Embed 5 Pro and Fast share an embedding space, enabling mixed-tier pipelines: index with Pro for quality and query with Fast for low latency.

Embed 5 accepts text, images, and mixed text-and-image inputs such as PDF pages for multimodal retrieval.

NEW INTEGRATION Cohere Embed 5 and Rerank 5 on OpenRouter

WAS THIS USEFUL? 👍 👎

IBM Granite 4.2 8B

Feedback ⓘ price increase · Nov 4, 2026

MODEL ↗

Granite 4.2 8B is a chat model from IBM.

Price increase on OpenRouter (ibm-granite/granite-4.2-8b): $0.06 → $0.06 in, $0.15 → $0.25 out per million tokens (+67%), first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

Meta Llama 3.1 70B Instruct

Feedback ⓘ

price increase · Nov 4, 2026 MODEL ↗

Llama 3.1 70B Instruct is a chat model from Meta.

Price increase on OpenRouter (meta-llama/llama-3.1-70b-instruct): $0.40 → $0.72 in, $0.40 → $0.72 out per million tokens (+80%), first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

Xiaomi MiMo-V2.5

Feedback ⓘ context change · Nov 4, 2026

MODEL ↗

MiMo-V2.5 is a multimodal model from Xiaomi.

Context window shrank from 1050K to 1M tokens on OpenRouter (xiaomi/mimo-v2.5), first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M2

Feedback ⓘ retirement scheduled · Nov 4, 2026 MODEL ↗

MiniMax M2 is a reasoning model from MiniMax.

OpenRouter lists a retirement date of 5 Nov 2026 for minimax/minimax-m2, first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M2-her

Feedback ⓘ retirement scheduled · Nov 4, 2026

MODEL ↗

MiniMax M2-her is a chat model from MiniMax.

OpenRouter lists a retirement date of 5 Nov 2026 for minimax/minimax-m2-her, first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M2.1

Feedback ⓘ retirement scheduled · Nov 4, 2026

MODEL ↗

MiniMax M2.1 is a reasoning model from MiniMax.

OpenRouter lists a retirement date of 5 Nov 2026 for minimax/minimax-m2.1, first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M2.7

Feedback ⓘ retirement scheduled · Nov 4, 2026

MODEL ↗

MiniMax M2.7 is a reasoning model from MiniMax.

OpenRouter lists a retirement date of 5 Nov 2026 for minimax/minimax-m2.7, first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

MiniMax-01 Feedback ⓘ retirement scheduled · Nov 4, 2026 MODEL ↗

MiniMax-01 is a multimodal model from MiniMax.

OpenRouter lists a retirement date of 5 Nov 2026 for minimax/minimax-01, first seen 4 Nov 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · November 5, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue