THE AI TOOLCHAIN

NO. 027

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 15, 2026 · EVERY DAY

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

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 249 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 41 tools

▾ // THE BRIEF HIDE

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

Three unrelated things stand out. Cline's RemoteEnvironmentService runs a full agent session on an SSH host while the SDK client stays , so the blast radius is the remote box. Braintrust turns recurring

vim ? help

trace patterns into automations rather than dashboards you have to read. And Claude Code adds per-command domain allowlisting to sandboxed sessions.

> BUILD
>
> **Run an agent session against a remote machine without shipping your client there**
>
> Cline's RemoteEnvironmentService keeps the SDK client local while the session executes over SSH, and Letta Code replaces its environments model with named remote computers; Devin Desktop adds Windows and macOS session platforms alongside Linux. The practical gain is that file writes and shell commands land somewhere disposable that already has the right toolchain, instead of on the developer laptop.
>
> Cline · Letta Code · Devin Desktop

> OPERATE
>
> **Act on a recurring failure in traces instead of finding it by scrolling**
>
> Braintrust's Loop-powered Patterns surface trends across traces and fire automations off them, which is the step between `we log everything` and `someone noticed`. mlxtop's client-side usage-file protocol gives exact token and latency numbers plus queue-depth and per-request prompt-load charts, so local serving slowdowns are attributable rather than felt.
>
> Braintrust · mlxtop

> GOVERN
>
> **Bound what an agent may reach before it reaches it**
>
> Claude Code adds per-command domain allowlisting and verified plugin installs to sandboxed sessions; Augment Code puts approval gates in front of sensitive files. OpenBot's `initiator` field lets a policy distinguish a scheduled run from a human session, so unattended automation can be held to stricter rules than an operator at a keyboard.
>
> Claude Code

BUILD

◆ AI Coding Agents

Augment Code

Feedback ⓘ 1 RELEASE · SEEN 2026-09-15 NOTES ↗

RANK

Your engineers have agents. Your organization doesn't.

> Augment Code tightened agent safety with approval gates on sensitive files, added proxy support for MCP servers, made a large sidecar heap increase configurable, and raised the minimum supported JetBrains IDE version.

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

01 **Configurable sidecar heap ceiling** IMPROVED 88

Adds the `-Daugment.sidecar.maxOldSpaceSizeMb` IDE system property to override the sidecar heap ceiling; the default is now 4096 MB, up from roughly 1.7 GB. Useful on machines loading very large conversations to avoid out-of-memory failures, e.g. `-Daugment.sidecar.maxOldSpaceSizeMb=8192`.

Increase the sidecar heap on a machine that loads very large conversations, to avoid out-of-memory failures.

env

```
-Daugment.sidecar.maxOldSpaceSizeMb=8192
```

— Names exact flag, old/new defaults, and a runnable override value. snapshot-20260915

02 **HTTP proxy routing for MCP servers** IMPROVED 66

MCP servers reached over HTTP, SSE, or Streamable HTTP now route through your configured HTTP proxy, e.g. by setting `HTTP_PROXY` and `HTTPS_PROXY` before launching.

Route MCP server traffic through a corporate proxy on a machine that requires one for outbound access.

```
$ export HTTP_PROXY=http://proxy.corp.example.com:8080
export HTTPS_PROXY=http://proxy.corp.example.com:8080
```

— Names three transport types and gives a working proxy env-var example.

snapshot-20260915

03 **Raised minimum JetBrains IDE version** BREAKING 65

The plugin now requires JetBrains IDEs 2025.2 (build 252) or newer; it previously installed on 2024.2 and later. Users on older IDEs will not receive further plugin updates.

— Gives exact old and new version requirements and consequence for laggards.

snapshot-20260915

THINNER COVERAGE BELOW

04 **Invalid Skill frontmatter surfaced in Settings** IMPROVED

48

Skills with invalid `SKILL.md` frontmatter are now surfaced in Settings with an actionable error and reason, instead of being silently dropped as before.

— Names the file and UI location but no exact fix steps. snapshot-20260915

05 **Approval gate on sensitive file reads** NEW

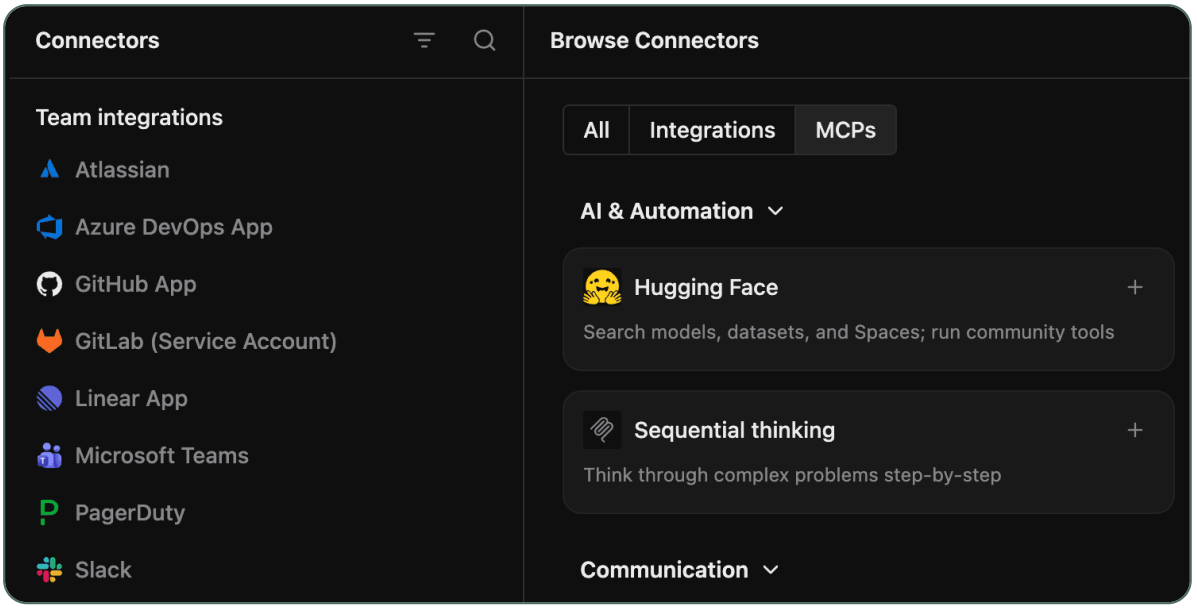
46

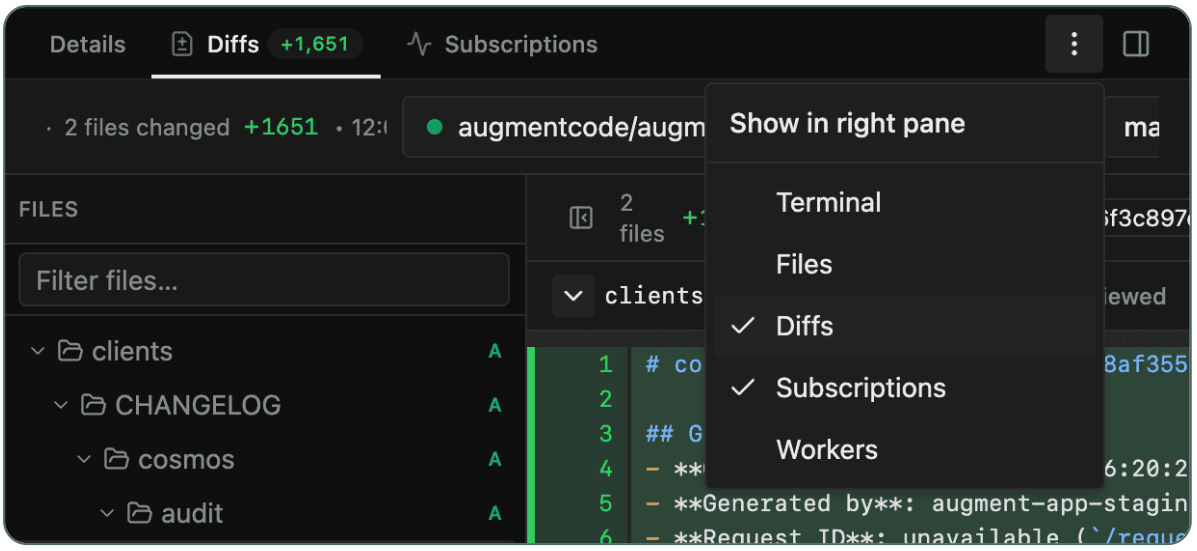
The Agent now prompts for approval before reading sensitive files, including `.env` files, private keys, certificates, credential- or secret-named files, and paths excluded by `.augmentignore`.

— Names the file categories and config exclusion but no command or setting to adjust it.

snapshot-20260915

└──▷ ALSO FROM THESE RELEASES





└──▷ **BREAKING ON UPGRADE**

! The plugin now requires JetBrains IDEs 2025.2 (build 252) or newer; it previously installed on 2024.2 and later. Users on IDEs older than 2025.2 will not receive further plugin updates.

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ 3 RELEASES · 2026-09-15 NOTES ↗ CODE ↗ RANK

Autonomous coding agent as an SDK, IDE extension, or CLI assistant.

> Cline's biggest addition this window is RemoteEnvironmentService, letting SDK clients run full sessions on a remote SSH host while staying local, alongside Hub-managed Agent Plugins, default-on provider-native web search, automatic retries for failed or mid-stream turns, three new model providers, and a breaking change to default models for 33 unpinned providers — plus a long tail of scheduling, session-import, checkpoint, and reliability fixes across the CLI, desktop app, and SDK.

└──▷ **WHAT SHIPPED · 23 FEATURES** most completely described first ? what's the number?

01 RemoteEnvironmentService runs sessions over SSH NEW

90

Adds `RemoteEnvironmentService` to run sessions on an SSH host while the client stays local: connect() uploads a helper binary to `~/.cline/remote/`, starts a loopback-bound Hub, and opens an SSH tunnel; the returned endpoint and token are passed to ClineCore.create({ backendMode: 'remote' }) so tools, sessions, and persistence all execute remotely. Hub daemons can also be started with connector management disabled so an SSH-spawned Hub does not adopt or restart the account's Slack and Telegram connectors.

— Names exact API calls and paths but no CLI walkthrough sdk/sdk/v0.0.83

02 Agent Plugins auto-discovered through the Hub NEW

90

Agent Plugins are now managed through the Hub: packages under `~/.agents/plugins/*` are auto-discovered and their skills are exposed as `plugin-name:skill-name` through the skills tool, and their MCP servers start without touching `cline_mcp_settings.json`.

Drop a plugin directory on the hub host to expose custom skills and MCP servers to all sessions without editing cline_mcp_settings.json.

```
$ mkdir -p ~/.agents/plugins/my-plugin/skills
cat > ~/.agents/plugins/my-plugin/plugin.json <<'EOF'
{ "name": "my-plugin", "version": "1.0.0" }
EOF
# Skills appear as my-plugin:<skill-name>; MCP servers listed in
mcp.json start automatically
```

— Names exact plugin path and includes a runnable setup example `cli-v3.0.62`

03 Provider-native web search enabled by default IMPROVED

80

Web search is now enabled by default in non-yolo sessions on provider/model combinations that support it, using provider-native search; an explicit `tools.web_search.enabled = false` still opts out, and the settings loader fails closed when a global settings file cannot be read.

Opt a specific environment out of provider-native web search when you want deterministic, search-free responses in CI.

```yaml
tools:
  web_search:
    enabled: false
```

— Names exact config key with a runnable opt-out example `sdk/sdk/v0.0.83` `cli-v3.0.62`

04 Default models change for 33 unpinned providers BREAKING

80

The resolved default model changes for 33 providers; any provider used without a pinned model will now send requests to a different model — most land on DeepSeek V4.1 Flash, NVIDIA moves to GLM 5.3 Flash, and NanoGPT moves to GPT Astra.

— Names exact provider migrations; readers should pin models `desktop-v0.0.28`

05 CLI startup notices for Cline Desktop and ClinePass NEW

75

A one-time CLI startup notice pointing at `cline.bot/desktop` now appears when the Cline Desktop app is introduced, with at most one notice per launch; the `CLINE_DISABLE_CLINE_PASS_NOTICE=1` environment variable now gates both this new Cline Desktop notice and the existing ClinePass intro notice.

Silence the new Cline Desktop startup notice (and all other startup notices) in CI or scripting environments where interactive prompts are unwanted.

```
$ CLINE_DISABLE_CLINE_PASS_NOTICE=1 cline "Run tests and fix any failures"
```

— Names exact env var and includes a runnable suppression command `cli-v3.0.62`

06 Automatic retry for failed provider turns IMPROVED

70

Failed turns caused by temporary provider errors are now retried automatically up to three times with backoff instead of immediately exiting with code 1; a turn that has already streamed any text, reasoning, media, or tool call is never retried. Model calls also now allow 5 SDK-level retries for request-start 429/5xx/network failures, up from 2.

– Explains mechanism and limits but nothing user-configurable `sdk/sdk/v0.0.83` `cli-v3.0.62` `desktop-v0.0.28`

---

07 **Scheduled and recurring run reliability fixes** IMPROVED

70

Scheduled runs now enforce parallelism limits at claim time inside a SQLite claim transaction, correctly resume work after system sleep, and decouple dispatch from execution so schedules no longer stall each other, with locally active claims renewed each tick; recurring schedules created without a timezone now persist the host's local IANA timezone instead of an implicit default.

– Detailed mechanism but no direct user action `sdk/sdk/v0.0.83` `cli-v3.0.62`

---

08 **Persistent snapshot index speeds up checkpoints** IMPROVED

70

Checkpoints now use a persistent per-session snapshot index so git's stat cache skips unchanged untracked files, reducing each snapshot after the second turn to about one git process; a corrupt index now heals with one rebuild-and-retry that also clears a stale `index.lock`.

– Explains mechanism and self-heal but not user-triggered `sdk/sdk/v0.0.83`

---

09 **PowerShell edition detection in command tool** IMPROVED

65

The command-running tool now reports which PowerShell edition is active — Windows PowerShell (`powershell.exe`) versus PowerShell (`pwsh.exe`) — and instructs the model to write commands directly rather than wrapping them in a second shell invocation, preventing pipeline corruption with `$_`.

– Internal fix; no direct user action described `desktop-v0.0.28`

---

10 **Imported sessions summarize foreign history on resume**

IMPROVED

65

Sessions imported from Claude Code, Codex, or opencode now summarize the foreign history on first resume rather than replaying tool calls the current agent cannot make; import origin is stamped on telemetry and resuming no longer overwrites stored import or automation provenance with a default `user` origin.

– Names source tools and behavior but no direct command `sdk/sdk/v0.0.83`

---

11 **Claude Code and OpenCode as local-auth providers** NEW

60

Claude Code and OpenCode are now declared local-auth providers with a named local CLI so hosts route them to a local-CLI readiness screen instead of demanding an API key.

— Names providers and behavior; UI path implied but not exact `sdk/sdk/v0.0.83`

THINNER COVERAGE BELOW

12 **Model picker offline fallback with full tiers** IMPROVED

55

Model pickers now fall back to generated Recommended, Free, and Subscribed tier lists (built at release time) when the models endpoint is unreachable, replacing the previous hand-maintained six-model offline fallback that had no subscribed tier.

— Describes fallback change but nothing to configure `sdk/sdk/v0.0.83` `cli-v3.0.62`

13 **Sign-in flow surfaces ClinePass plan details** IMPROVED

55

The sign-in flow with a Cline account now surfaces plan details — free model promotions, ClinePass coverage across DeepSeek, Kimi, and GLM, and the absence of an API key requirement — on the sign-in card and final onboarding step.

— UI copy change, no actionable surface named `desktop-v0.0.28`

14 **Streaming no longer throttled by hook forwarding** IMPROVED

55

Per-chunk text, reasoning, and tool-update deltas are no longer forwarded to remote `onEvent` hooks, and the hub event log moves to `synchronous = NORMAL`, removing a source of streaming throttling.

— Names setting change but no user-facing action `sdk/sdk/v0.0.83`

15 **File index skipped for home or root workspaces** IMPROVED

55

Skips building a file index when the workspace root is the home directory or a filesystem root, preventing an OOM-kill when typing `@` mentions from `$HOME`.

— Explains bug and trigger but no config surface `sdk/sdk/v0.0.83`

16 **Model picker refreshes on open** IMPROVED

50

The model picker now refreshes its provider and model list every time it is opened, including live Recommended and Free groupings, rather than freezing until app restart.

— Describes UI behavior; no config surface named `desktop-v0.0.28`

17 **Three new model providers added to catalog** NEW 50

Adds three new model providers to the catalog: Infer by Flow7, Melious, and Wallaby, expanding the catalog from 5,923 to 6,079 models.

— Names providers and counts but no usage steps `desktop-v0.0.28`

18 **Langfuse tracing limited to Cline providers** IMPROVED 50

Langfuse tracing is now limited to the Cline and Cline Pass providers, preventing prompts and responses from third-party and BYOK providers from being exported in Langfuse-configured environments.

— Names affected providers, no toggle described `sdk/sdk/v0.0.83`

19 **Session history listing fixes hidden root sessions** IMPROVED 50

Session history listing now widens its fetch window until the requested page fills and filters child rows in SQL, so root sessions are no longer hidden behind their children.

— Explains fix but no interface detail given `sdk/sdk/v0.0.83`

20 **ClinePass subscribers default to subscribed-tier model** IMPROVED 45

ClinePass subscribers are now defaulted to a model from their subscription tier rather than the most recently published free model, across both the CLI and desktop app.

— Default change explained but no config option given `cli-v3.0.62` `desktop-v0.0.28`

21 **Sessions get default system prompt when unset** IMPROVED 45

A session started without an explicit `systemPrompt` now gets a default Cline prompt built on the execution host rather than running with none.

— Names field but gives no usage example `sdk/sdk/v0.0.83`

22 **Automation event acceptance is now atomic** IMPROVED 40

Automation event acceptance now runs in one write transaction; failures roll back and propagate to the caller for redelivery.

— Brief mechanism description, no user action `sdk/sdk/v0.0.83`

23 **Credential fields sanitized before saving** IMPROVED 35

Credential fields are now stripped of Unicode control and format characters and surrounding whitespace before being saved.

— Security fix with minimal scope detail `sdk/sdk/v0.0.83`

└──▷ BREAKING ON UPGRADE

! The resolved default model changes for 33 providers; any provider used without a pinned model will now send requests to a different model (most land on DeepSeek V4.1 Flash, NVIDIA moves to GLM 5.3 Flash, NanoGPT moves to GPT Astra).

WAS THIS USEFUL?  

Red Hat ripwire

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗

RANK

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

> ripwire's v0.6.1 window adds directory-scoped churn queries, scip-java index ingestion, native Elixir resolution, a self-contained HTML call-graph visualizer, and unproven-definition reporting across query verbs, alongside several breaking changes to cache format, `--for` semantics, and file symlink handling.

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

01 Directory-scoped churn queries via `--in=DIR` NEW

95

The `--in=DIR` flag, used with `--rank-by=churn-decay`, scopes what changed recently queries to a single directory, collapsing the full repository map into a stub with a scoped `<recent scope= n= of= merge_bombs_skipped=>` block — on RocksDB this shrinks an answer from 39.8 KB to 10.2 KB per call. `--in=DIR` is refused with any verb other than `--rank-by=churn-decay`, and with multi-root, `--top-k=0`, or `--json`.

Scope a what changed recently query to a subdirectory to get a dramatically smaller answer — useful in large monorepos where you only care about one component.

```
$ ripwire --in=src/storage --impact
```

Scope a what changed recently question to a single directory to get a tight, directory-local churn answer instead of a full repository map.

```
$ ripwire . --rank-by=churn-decay --in=src/db
```

— Exact flag, output block format, sizes, and refusal conditions. v0.6.1

02 `--for=TASK --limit=N` now returns file-grain widening page

95

BREAKING

`--for=TASK --limit=N` no longer behaves as in 0.6.0: it now serves a file-grain widening page with one `<f p= score= n= sym=/>` row per positive-score file, paged with `--offset=M`. Bundle-shaping flags alongside it are now refused rather than ignored, and `--top-k` stays inert on `--for`.

Widen a task answer to a file-by-file page when you want to see which files are most relevant before drilling into symbols.

```
$ ripwire . --for='add connection pooling to the storage layer' --limit=20 --offset=0
```

— Exact before/after behaviour, flags, and a runnable example. v0.6.1

03 **Symlink protection on notes and baseline files** BREAKING 90

`.ripwire_notes`, `.ripwire_quality_baseline`, and `.ripwire_arch_baseline` are now opened with `O_NOFOLLOW`; a symlink at any of those names is refused on both read and write. Replace any symlink with a regular copy of its target, or every read prints a refusal on stderr, `--quality-delta` compares against HEAD, `--arch` reports every violation as new, and `--note-add`, `--quality-baseline`, `--arch --baseline`, and `--baseline-update` exit 1 without writing.

— Exact filenames, syscall, and remedy for every affected command. v0.6.1

04 **Self-contained HTML call-graph visualization** NEW 88

`--html[=FILE]` emits a self-contained HTML call-graph visualization with no server or CDN dependency. `--color-by=lang|community|cx|churn|tested` embeds all five colour lenses in one file with a live selector to switch between them.

Visualise the call graph coloured by churn to spot high-risk, frequently-changing code before a review.

```
$ ripwire . --html=graph.html --color-by=churn
```

— Exact flags, output format, and a runnable example. product docs

05 **scip-java index ingestion via `--scip`** NEW 86

The `--scip` flag lets ripwire read scip-java indexes to map callers and tests without a live language server. Support was extended to the typed range encoding scip-java writes, raising precise overlay matches from 0% to 79% of occurrences on spring-petclinic.

Ingest a scip-java index to map callers and tests for a Java codebase without running a live language server.

```
$ ripwire --scip=index.scip --callers
```

— Named flag with a concrete before/after percentage and example. v0.6.1

06 **Unproven-definition reporting across query verbs** NEW

80

Adds `unproven_defs=` output to `--callers`, `--callees`, `--impact`, `--safe-delete`, `--path`, `--uses`, `--mentions`, `--verify`, `--affected`, and `--edit-check` when a header selector drops definitions it cannot prove, replacing the previous silent zero with an explicit reported count.

Check callers of a C++ header symbol and see how many definitions could not be proven, rather than receiving a quiet zero.

```
$ ripwire . --callers=src/storage/Store.h:putObject --legend=full
```

— Names every affected verb and the exact output field. v0.6.1

07 **`--legend=compact` default for generated agent commands**

IMPROVED

78

Generated agent commands — skills, the `ripwire wrap` paste block, prompt routers, and tool routes — now carry `--legend=compact` by default; add `--legend=full` to retrieve full definitions. The bare CLI default is unchanged.

— Exact flag, default behaviour, and override are all named. v0.6.1

08 **Native Elixir resolution by module, name, and arity** NEW

73

ripwire resolves Elixir code natively by module, name, and arity, covering lexical aliases, filtered imports, default arguments, pipes, captures, delegates, nested modules, `defimpl` multi-targets, types, and callbacks.

— Detailed mechanism list but no runnable example given. v0.6.1

09 **Output size and performance reductions** IMPROVED

70

Compact answers shrink 46–66% (2.8–5.8 KB less per call) on `--callers`, `--uses`, `--impact`, and `--affected`, with the flagless map 15.3% smaller at identical rows. The declined-call index memory footprint drops from 114 MB to 368 KB on llvm-project by storing one copy per distinct candidate list, with every count and byte of output unchanged, and cold parse time on llvm-project (182,555 files) falls from 194 s to 156 s of CPU.

— Concrete before/after numbers but nothing new to invoke. v0.6.1

10 **Cache format bump for C/C++ internal linkage** BREAKING

65

Cache format moves from 21 to 22 (and parser version to 96) to record the internal-linkage bit on every C and C++ definition; a cache written by 0.6.0 is rejected and the tree is rebuilt automatically on the first run.

— Exact version numbers, though rebuild requires no user action. v0.6.1

11 **Grammar and Ruby resolution expansion** IMPROVED 63

Adds Kotlin and Dart vendored grammars, bringing the total to 24. Ruby support was expanded to read superclasses, mixins, `autoload`, and constant receivers that a Rails autoloader loads through, and the Ruby dependency view now also recognises a constant argument and a rescue class as dependencies.

— Names every added grammar and Ruby construct, no example given. v0.6.1

THINNER COVERAGE BELOW

12 **Definitions for compact-output fields** IMPROVED 58

Every number in a compact answer now arrives with its definition, including previously undefined fields `graph_unindexed=`, `declined_calls=`, `unproven_defs=`, `pr_iters=`, `--impact` blast-radius counts, `--safe-delete` verdict fields, and `--communities` / `--community` structure counts.

— Lists exact fields but is a documentation change, not a mechanism. v0.6.1

13 **`sc=` replaces `id=` in symbol row output** BREAKING 35

`sc=` replaces `id=` on map and lens symbol rows in output.

— Bare rename noted with no migration mechanism given. v0.6.1

└──▷ BREAKING ON UPGRADE

! Cache format moves from 21 to 22 (and parser version to 96) for the internal-linkage bit on every C and C++ definition; a cache written by 0.6.0 is rejected and the tree is rebuilt on the first run.

! `.ripwire_notes`, `.ripwire_quality_baseline`, and `.ripwire_arch_baseline` are now opened with `O_NOFOLLOW`; a symlink at any of those names is refused on both read and write. Replace any symlink with a regular copy of its target, or every read prints a refusal on stderr, `--quality-delta` compares against HEAD, `--arch` reports every violation as new, and `--note-add`, `--quality-baseline`, `--arch --baseline`, and `--baseline-update` exit 1 without writing.

! `--for=TASK --limit=N` no longer means what it meant in 0.6.0: it now serves a file-grain widening page (`--offset=M` paging) rather than the previous behaviour. `--top-k` stays inert on `--for`.

! `--in=DIR` is refused (naming the remedy) with any verb other than `--rank-by=churn-decay`, with multi-root, with `--top-k=0`, and with `--json`.

! `sc=` replaces `id=` on map and lens symbol rows in output.

WAS THIS USEFUL? 👍 👎

DeepSeek Harness

Feedback ⓘ 1 RELEASE · 2026-09-15 CODE ↗

RANK

DeepSeek Harness is an open-source agent harness for building coding agents with plugins.

> DeepSeek Harness's v0.1.6-alpha.1 release adds computer-use support via Cua Driver, upgrades MCP protocol handling with scoped resources, and tightens control over experimental package distribbution.

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

01 **Computer-use via Cua Driver providers** NEW 40

Adds computer-use capability through Cua Driver providers, letting the harness drive computer-use style interactions as part of an agent's toolset.

— Names the driver but no config or usage detail given. dsh-v0.1.6-alpha.1

02 **MCP protocol negotiation and scoped resources** NEW 38

Negotiates modern MCP protocols using the official SDK, adds scoped MCP resources and server instructions, and activates profile resource tools for configured MCP servers.

— Describes MCP surface changes without config keys or commands. dsh-v0.1.6-alpha.1

03 **Experimental package denylist and opt-in publishing** NEW 33

Uses a private denylist to keep experimental packages out of default product releases, while publishing all experimental packages as a separate opt-in set.

— Explains mechanism but no named flags or config for opting in. dsh-v0.1.6-alpha.1

04 **Maintained repository reference policy** IMPROVED 18

Enforces a policy requiring references to point at maintained repositories.

— Bare statement with no mechanism or scope. dsh-v0.1.6-alpha.1

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗

CODE ↗ RANK

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

> Codex CLI's rust-v0.155.0-alpha.6 release adds opt-in registered package execution for the Windows sandbox, new daemon PID tracking and ownership diagnostics, a separate daemon package for smoother updates, and early attachment upload APIs.

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

01 Registered package execution in Windows sandbox NEW

85

Adds `CODEX_WINDOWS_REGISTERED_CORE=1` environment variable to opt in to registered package execution in the Windows sandbox, launching runners through service-recorded execution aliases with ownership and OS package identity validation.

Enable registered package execution in the Windows sandbox so Codex launches sandboxed runners through validated OS execution aliases instead of copied helpers.

```
$ CODEX_WINDOWS_REGISTERED_CORE=1 codex
```

— Named env var, mechanism explained, and runnable example given `rust-v0.155.0-alpha.6`

02 Daemon PID tracking and package ownership diagnostics NEW

70

Adds dedicated `daemon.pid` and `daemon-updater.pid` files in the state directory, surfaced as diagnostics in `codex app-server daemon` health checks. Also adds a `--check-package-ownership` flag to the internal `app-server daemon pid-update-loop` subcommand to probe daemon-owned package support without starting the updater.

— Names exact files, subcommand and flag but no usage example `rust-v0.155.0-alpha.6`

THINNER COVERAGE BELOW

03 Separate daemon package for CLI updates NEW

40

Adds a dedicated daemon package separate from the standalone CLI installation, allowing daemon updates to preserve the user's CLI selection and shell profile.

— Describes benefit but no config key or command named `rust-v0.155.0-alpha.6`

04 **Attachment upload and resolution APIs** NEW 20

Adds attachment upload and resolution APIs and passes stores into sessions.

— Bare mention with no endpoint names or usage shown `rust-v0.155.0-alpha.6`

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-08-16 CODE ↗

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

Adds treemap, waterfall, dumbbell, slopegraph, ridgeline, bubble, and ten editorial diagram types, plus Excalidraw import and native Droid plugin packaging.

Adds `treemap` chart type for part-of-whole visualisation by area.

Adds `dumbbell` as a variant of the bar chart type.

Adds `slopegraph` as a Line variant for showing change between two states.

Adds `ridgeline` as a Line variant.

Adds `bubble` as a scatter variant for three-value comparisons.

+5 more

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ RANK

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

> Letta Code replaces its environments model with named remote computers, adds an external skills system with three memory scopes, and rounds out automation with hooks, crons, permissions, secrets, subagents, a desktop app, and Git-backed memory sync — while removing AgentFile export/import entirely.

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

01 **Skill installation and management commands** NEW 95

Adds `letta skills install <url>` to install skills from GitHub URLs or third-party hubs (ClawHub, Hermes Skills Hub) into an agent's memory, `letta skills list --agent <agent>` to list them, and `letta skills delete <skill> --agent <agent>` to remove them, alongside `/skills` to view loaded skills and `/skill-creator` to create new ones. Skills load from three scopes: global (`~/.letta`), project-scoped (`.agents/skills`), and agent-scoped (stored in MemFS).

Install a community skill from GitHub directly into a specific agent's memory.

```
$ letta skills install
https://github.com/owner/repo/blob/main/path/to/skill/SKILL.md
```

— Full command set and scope mechanics are named. product docs

02 Remote computers replace environments for agent targeting

90

BREAKING

Adds `letta server --computer-name "..."` to register a machine as a named remote computer and a `--computer <name>` flag to route agent messages to it, with `--computer cloud` targeting the agent's cloud sandbox. This replaces the environments model: the `environments` / `envs` subcommands and `--environment` / `--env` / `--env-name` flags have been removed.

Register a remote machine so agents can be dispatched to it from any other computer.

```
$ letta server --computer-name "work-laptop"
```

— Exact commands and flags for both the new model and the removal. product docs

03 Sync agent memory to a GitHub repository NEW

80

The `/memory-repository set git@github.com:...` slash command syncs all agent context (MemFS, memory blocks) to a custom GitHub repository.

Sync an agent's full context (memory blocks, MemFS) to a private GitHub repo for version-controlled backup.

```
$ /memory-repository set git@github.com:myorg/my-agent-memory
```

— Exact command shown, mechanism is a bit thin. product docs

04 New slash commands for memory, search, and model config NEW

75

Adds `/sleeptime` to configure periodic dreaming (sleep-time compute) for continuous self-improvement, `/palace` to view agent memory, `/search` to search across all messages and agents, `/connect` to configure LLM API keys (OpenAI, Anthropic, Z.ai, etc.), `/model` to swap active models, and `/login` (plus desktop app sign-in) to access agents stored in Letta Cloud.

— Six named commands, each described only in one line. product docs

05 AgentFile export and import removed BREAKING

75

AgentFile (`.af`) export and import have been removed; the `/export` and `/download` slash commands and the `--import` and `--from-af` CLI flags are no longer supported, including imports from the agent registry.

— Names every removed command and flag exactly. product docs

06 **Subagents and multi-agent support** NEW 60

Adds Subagents and multi-agent support, including built-in general-purpose, forked, recall, and history-analyzer subagents callable in the background.

— Names subagent types but no invocation syntax. product docs

THINNER COVERAGE BELOW

07 **Crons and schedules for agent automation** NEW 50

Adds Crons and Schedules support for heartbeats, cron-based triggers, and self-managed agent schedules.

— Names trigger types but no commands or config. product docs

08 **Secrets management across machines** NEW 50

Adds Secrets management to expose secrets as environment variables across machines while obfuscating their values from agent context.

— Mechanism described but no config keys or commands. product docs

09 **Messaging integrations for Telegram, Slack, and Discord** NEW 45

Adds messaging integrations with Telegram, Slack, and Discord, plus support for custom channels.

— Named integrations but no setup mechanism given. product docs

10 **Permissions system for agent actions** NEW 45

Adds a Permissions system to set permission modes and customize which agent actions are auto-approved or auto-denied.

— Describes behavior but no config surface named. product docs

11 **Desktop app for macOS, Windows, and Linux** NEW 45

Adds a desktop app for macOS, Windows, and Linux that shares agents with the CLI.
.

— Platforms named, no further detail on setup. product docs

12 **Hooks for custom scripts during agent execution** NEW 35

Adds Hooks support to run custom scripts at key points of agent execution for workflow automation.

— No hook points, config, or syntax specified. product docs

13 **Curated agent name generator** IMPROVED 20

Curates memorable agent names drawn from Fallout, Dune, and sci-fi classics.

— Cosmetic change with no mechanism or usage detail. v0.32.9

└──▷ BREAKING ON UPGRADE

! AgentFile (`.af`) export and import have been removed; the `/export` and `/download` slash commands and the `--import` and `--from-af` CLI flags are no longer supported, including imports from the agent registry.

! The `environments` / `envs` subcommands and `--environment` / `--env` / `--env-name` flags have been removed (replaced by the remote computers model).

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ 1 RELEASE · 2026-09-14

NOTES ↗ CODE ↗ RANK

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

> Claude Code v2.1.271 adds fine-grained sandboxing and provisioning controls for agentic sessions, including per-command domain allowlisting, subagent CLAUDE.md opt-out, verified plugin installs, and fast mode for Remote sessions.

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

01 Verified plugin install/update via command hash NEW

90

Adds `--accept-command <sha256>` to `claude plugin install` and `claude plugin update`, letting an operator accept exactly the command a prior `--json` run displayed instead of approving broadly with `-y`. This lets CI pipelines run a dry-run with `--json`, review the exact command, then re-run with the matching hash for unattended installs.

Automate unattended plugin installs in CI by accepting the exact command hash surfaced by a prior dry-run, avoiding the broad `-y` approval

```
$ claude plugin install <plugin-dir> --json 2>&1 | tee
install.json
# then, after reviewing the displayed command:
claude plugin install <plugin-dir> --accept-command <sha256>
```

— Named flags plus a runnable before/after CI example v2.1.271

02 Graceful drain signaling for self-hosted runners NEW

85

The `--drain-marker-file` flag on `claude self-hosted-runner` lets a runner receiving SIGTERM signal a graceful drain rather than a crash, so the server records the correct exit reason.

Signal a graceful host drain (rather than a crash) when a self-hosted runner receives SIGTERM, so the server records the correct exit reason

```
$ claude self-hosted-runner --drain-marker-file
/run/claude/drain.marker
```

— Named flag with runnable command, but not described in release notes v2.1.271

03 **Per-command domain sandboxing in auto mode** NEW 83

Adds per-command `allowed_domains` to Bash, PowerShell and Monitor in auto mode: the hosts a command needs are reviewed alongside it and opened for that command alone, while all other hosts are refused.

— Named config key with clear allow/deny mechanism, no example v2.1.271

04 **Marked-up pricing multiplier for chargeback** NEW 78

Adds support for a `multiplier` value above 1 (up to 10) in the `modelPricing` managed setting and the Claude apps gateway `pricing` block, enabling marked-up internal chargeback rates.

— Named config keys and numeric limit, no runnable example v2.1.271

05 **Fast mode for Remote sessions** NEW 72

Adds fast mode to Claude Code Remote sessions on cloud and self-hosted runners: the host's fast-mode setting, or `/fast` typed in the session, applies where the organization allows it.

— Names a typed command trigger but no worked example v2.1.271

06 **CLAUDE.md opt-out for subagents** NEW 70

Adds `omitClaudeMd` to agent frontmatter and to the `--agents` JSON input, letting custom and plugin subagents run without loading user, project and local CLAUDE.md files, while managed policy files still load.

— Named config key and scope, no usage example v2.1.271

THINNER COVERAGE BELOW

07 **Mouse support in /config panel** IMPROVED 55

Adds mouse support to the `/config` panel in fullscreen mode: the scroll wheel moves the settings list, a click on a value changes it, and the row under the pointer is highlighted.

— Clear UI behaviour, only a navigation path to act on v2.1.271

WAS THIS USEFUL?  

SST OpenCode

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ CODE ↗

RANK

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

> OpenCode's v1.18.31 release adds batch workspace block/unblock API endpoints for support operations and improves GitHub Copilot's adaptive thinking summarization to match VS Code behavior.

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

01 **Batch workspace block/unblock endpoints** NEW 90

New `POST /api/support/actions/block-workspaces` and `POST /api/support/actions/unblock-workspaces` endpoints accept a `{ "workspaceIDs": [...] }` body with `SUPPORT_API_KEY` bearer auth, letting an operator block or unblock arbitrary lists of workspace IDs in a single request. Each call returns per-item `{ blocked | unblocked, notFound }` results instead of requiring one call per workspace.

Block hundreds of compromised workspaces in one round trip during a fraud response, instead of calling a single-workspace endpoint per ID.

```
$ curl -X POST https://<host>/api/support/actions/block-
workspaces \
  -H 'Authorization: Bearer <SUPPORT_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"workspaceIDs": ["wrk_aaa", "wrk_bbb", "wrk_ccc"]}'
```

Roll back a mistaken batch block by unblocking the same workspace IDs in one call.

```
$ curl -X POST https://<host>/api/support/actions/unblock-
workspaces \
  -H 'Authorization: Bearer <SUPPORT_API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"workspaceIDs": ["wrk_aaa", "wrk_bbb", "wrk_ccc"]}'
```

— Full endpoint paths, auth, request/response shape, and runnable curl examples. v1.18.31

THINNER COVERAGE BELOW

02 **Summarized adaptive thinking for GitHub Copilot** IMPROVED 44

Requests now use the `adaptive_thinking` capability with `display: 'summarized'` for GitHub Copilot models, matching VS Code's behavior for summarizing model thinking output.

— Names the capability and setting but no usage example or scope. v1.18.31

WAS THIS USEFUL?  

Alibaba Qwen Code

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗

RANK

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

> Qwen Code's biggest changes this window center on Web Shell, which gained preview panels, inline message editing, and a batch of session-management improvements, alongside new Goal turn/time limits, a Codex subagent executor, an agent board for sharing work across agents, and two breaking changes to channel message filtering and hook timeouts.

└──▷ WHAT SHIPPED · 18 FEATURES most completely described first ? what's the number?

01 **Goal turn and time limits** NEW 80

Adds `model.goalMaxTurns` and `model.goalMaxActiveMinutes` config keys to cap how long a Goal can run, with active time rebased on session restore.

Cap a long-running Goal to 20 turns and 30 minutes so it does not consume unbounded resources in CI.

```json
{
  "model": {
    "goalMaxTurns": 20,
    "goalMaxActiveMinutes": 30
  }
}
```

— Named config keys with a runnable example v0.23.4

02 **Effort and tool restriction options for agent()** NEW 75

Adds `effort` and `disallowedTools` options to the agent() workflow function, letting a workflow set a subagent's effort level and block specific tools.

Give a subagent an effort level and block specific tools when orchestrating from a workflow.

```javascript
agent("Audit the auth module for injection risks", {
  effort: "high",
  disallowedTools: ["Bash", "WriteFile"]
})
```

— Named function options with a runnable example v0.23.4

03 **Web Shell add-source URL parameter** IMPROVED 70

Adds the `addSource=1` URL parameter to re-enable the Web Shell sources panel add-source button, which is now hidden by default.

Expose the Web Shell add-source button for a session where you need to attach additional sources.

```
$ qwen serve --open
# Then navigate to the Web Shell URL with ?addSource=1 appended
```

— Named URL parameter with a shell example v0.23.4

04 **Web preview panels in Web Shell** NEW 70

Adds Web preview panels in Web Shell with desktop and mobile widths, refresh, external opening, saved HTML artifacts, and per-session URL and width memory.

— Rich feature list but no exact command or path v0.23.4

05 **Web Shell session and interaction improvements** IMPROVED 70

Adds inline editing and resending of the latest user message in Web Shell, retaining attached images and files; a Continue execution action for recoverable interrupted sessions; model role and context window configuration; manual context compression in the context overview and composer context hover; automatic start of approved Goal proposals after their owning turn finishes, removing the need to copy a manual `/goal set` command; scheduled runs that route to a configured model and session group when New session each run is selected; browser notifications that identify the session, show prompt and reply excerpts, and open the target session when clicked; and navigable Session Workflow dependencies.

— Names each Web Shell increment and the `/goal set` command v0.23.4

THINNER COVERAGE BELOW

06 **Read hook timeout unit changed to seconds** BREAKING 55

The `read` command hook timeout is now interpreted in seconds (previously milliseconds).

— Exact before/after unit change, actionable to update configs v0.23.4

07 **Message-prefix filtering removed from channels**

BREAKING

50

Configurable message-prefix filtering is removed from channels; eligible messages now follow normal sender, group, mention, and pairing policies without a prefix.

— Explains before/after but no migration steps v0.23.4

08 **Codex subagent executor and CodeModeOnly tool mode** NEW

45

Adds a Codex executor and native agent builtins for subagents, plus a new `CodeModeOnly` tool execution mode.

— Names the mode but no behavior detail v0.23.4

09 **New fields in hook input payloads** NEW

40

Adds `permission_mode`, `agent_id`, and `prompt_id` fields to every hook input payload.

— Names fields but shows no usage v0.23.4

10 **Configurable web search budget** NEW

40

Adds a configurable `web_search` budget and bounds the extractor fallback.

— Names the config but gives no values or usage v0.23.4

11 **Peer SDK endpoint for cross-session messaging** NEW

40

Adds a peer endpoint so a program outside Qwen Code can join cross-session messaging.

— No endpoint path or protocol given v0.23.4

12 **Visual input and local memory in Qwen Live** NEW

40

Adds visual input, proactive assistance, and local memory to the standalone Qwen Live experience.

— Lists capabilities without mechanism or limits v0.23.4

13 **App-bound actions for computer-use agent** NEW

35

Adds app-bound actions and compact native observations to the computer-use agent.

— No specifics on which apps or observations v0.23.4

14 **Stalled Goal checkpoint retry batching** IMPROVED

35

Stalled Goal checkpoints now rerun in batches, with smaller batches after each stall.

— Describes behavior but no batch sizes v0.23.4

15 **Agent board for cross-agent work sharing** NEW 30

Adds an agent board feature for sharing work across independently started agents.

— No UI path or mechanism given v0.23.4

16 **Explicit session source switch in VS Code** NEW 30

Adds an explicit session source switch in VS Code.

— No detail on options or UI location v0.23.4

17 **Auto-retry on transient network errors** IMPROVED 30

Auto-retries transient network errors (EOF) in core where Ctrl+Y is unavailable.

— Narrow fix with minimal mechanism detail v0.23.4

18 **Extension skills named by their extension** IMPROVED 25

Extension skills are now named by their extension.

— One-line change with no further detail v0.23.4

└──▷ **BREAKING ON UPGRADE**

! Configurable message-prefix filtering is removed from channels; eligible messages now follow normal sender, group, mention, and pairing policies without a prefix.

! The `read` command hook timeout is now interpreted in seconds (previously milliseconds).

WAS THIS USEFUL?  

Amazon Kiro

Feedback ⓘ 2 RELEASES · 2026-09-14 BLOG ↗ RANK

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

> Kiro's biggest changes this window were a 1M-token context window (up from 272K) for the GPT-5.6 Sol, Terra, and Luna models with new two-tier billing, and the Kiro 1.1 IDE release adding durable agent artifacts, native ARM64 builds, and enterprise profile region routing.

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

01 **1M context window for GPT-5.6 models** IMPROVED 85

Expands the context window for GPT-5.6 Sol, Terra, and Luna from `272K` to `1M` tokens in the Kiro IDE, CLI, and Web, enabling repository-scale refactors, end-to-end debugging, and long-horizon agentic tasks without chunking. Introduces a two-tier credit billing model: requests up to 272K tokens bill at the short-context rate (Sol 4.4x, Terra 2.2x, Luna 1.1x), while requests above 272K tokens bill at double that rate.

Activate the updated 1M-context GPT-5.6 models after the rollout by restarting the Kiro CLI so your next long-horizon agentic task picks up the expanded window.

📍 Restart your IDE or CLI, or refresh Kiro Web, to see the updated models.

— Names exact token limits and per-model billing multipliers

Models: GPT-5.6 Sol, Terra, and Luna up…

02 **Durable agent artifacts in Agent Focus** NEW 65

Agents can now produce durable artifacts from documents, diagrams, code, and images, reviewable inline as a compact preview or expanded in full beside the chat in Agent Focus, and accessible later from the Agent Focus context panel.

Ask the agent to produce a design doc as a durable artifact you can review and return to, rather than hunting through chat history.

📍 In the Agent Focus chat, prompt Kiro to generate a document or diagram. When the artifact appears, click the compact preview to expand it beside the chat, or reopen it later from the Agent Focus context panel.

— Describes behaviour and exact UI navigation to use it

IDE: Agent Artifacts and Native ARM64 B…

THINNER COVERAGE BELOW

03 **Native ARM64 builds for Windows and Linux** NEW 50

Adds native ARM64 builds for Windows and Linux, available from the Kiro downloads page, eliminating the need for x64 emulation.

— Names platforms and download location but no further detail

IDE: Agent Artifacts and Native ARM64 B…

04 **Enterprise profile region routing** IMPROVED 50

Enterprise profiles now route service requests to the selected profile's region when signing in with an enterprise identity, even when the identity provider is registered elsewhere.

— Explains mechanism but no config surface to act on

IDE: Agent Artifacts and Native ARM64 B…

05 **Windows enterprise policy path moved to Kiro registry** 50

BREAKING

Kiro now reads managed Windows enterprise policies from the Kiro registry path instead of the VS Code policy path; policies stored only under the VS Code path will no longer be applied.

— Clear before/after but exact registry path not given

IDE: Agent Artifacts and Native ARM64 B…

06 **Editor foundation updated to Code OSS 1.131** IMPROVED 45

Updates the editor foundation from Code OSS `1.109.5` to Code OSS `1.131`, bringing 22 platform releases of editor, terminal, accessibility, and extension-host updates.

— Names exact version jump but no specific changes listed

IDE: Agent Artifacts and Native ARM64 B…

07 **MCP tool health and permission defaults** IMPROVED 45

MCP servers that cannot list their tools now appear as failed instead of connected with zero tools, and MCP tool annotations inform default permission decisions.

— Describes two related MCP fixes without a config surface

IDE: Agent Artifacts and Native ARM64 B…

08 **CONTRIBUTING.md convention detection** NEW 40

Kiro follows repository `CONTRIBUTING.md` instructions when they define branch, commit, or pull request conventions.

— Names the exact file but no mechanism detail IDE: Agent Artifacts and Native ARM64 B…

09 **Agent Focus window and checkpoint reliability fixes** IMPROVED

30

Agent Focus reopens with the same window style and size after a restart, and checkpoint history continues paging correctly after a revert.

— Two minor bug fixes described only briefly IDE: Agent Artifacts and Native ARM64 B…

└──▷ **BREAKING ON UPGRADE**

! Managed Windows enterprise policies are now read from the Kiro registry path instead of the VS Code policy path — policies stored only under the VS Code path will no longer be applied.

WAS THIS USEFUL? 👍 👎

Cotool

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ RANK

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

> Cotool v0.69.0 adds a LaunchDarkly integration for searching audit logs, broadens agent failure alerting, and adds setup guidance for connecting Microsoft Sentinel.

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

01 **LaunchDarkly audit log search integration** NEW 38

A new LaunchDarkly integration lets teams search account audit logs from within Cotool.

— Names the integration but no config or endpoint detail given. v0.69.0

02 **Azure setup guidance for Microsoft Sentinel** NEW 35

Adds step-by-step Azure setup guidance for connecting Microsoft Sentinel to Cotool.

— Names the integration and setup guidance but no exact steps included. v0.69.0

03 **Expanded agent failure notifications** IMPROVED 31

Agent failure notifications now alert configured destinations whenever a run fails, expanding on prior notification coverage.

— Describes behavior but not which destinations or how to configure them. v0.69.0

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

CopilotKit

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ RANK

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

> CopilotKit v1.71.2 adds native Intelligence runtime support by integrating the MCP Apps middleware into its TypeScript runtime, with tool-visibility scoping, AG-UI schema validation, and an MCP MIME type correction.

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

01 **MCP Apps middleware integration in runtime** NEW 80

Integrates `@ag-ui/mcp-apps-middleware@^0.1.0` into `@copilotkit/runtime`, running the TypeScript runtime against the same cross-language conformance suite as CopilotKit's native runtimes. It enforces MCP tool visibility filtering and unambiguous account/proxy selection, rejecting proxy requests outside the selected agent scope, and validates AG-UI tool arguments against their JSON schemas, keeping relaxed structured output limited to the A2UI tool. It also advertises and consumes the January MCP Apps MIME type correction (`text/html;profile=mcp-app`).

— Names exact package version, MIME type and validation scope; no usage example given

v1.71.2

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ RANK

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

> AI SDK 7.0.100 adds typed error exposure for UI transport and completion failures, enabling more structured error handling in client applications.

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

01 **Typed errors for UI transport and completion** IMPROVED

35

Exposes typed AI SDK errors for UI transport and completion failures, enabling structured error handling in client code.

– States the change but no error type names or handling example given. ai@7.0.100

WAS THIS USEFUL? 👍 👎

CopilotKit OpenBot

Feedback ⓘ 1 RELEASE · 2026-09-14

NOTES ↗ RANK

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

> OpenBot's v0.0.11 release adds an `initiator` field to policy context so boundary rules can tell scheduled runs from human sessions, refines credential-content detection and audit/routines UI reporting, and brings refusal-reason forwarding to the Python LangGraph Bot; a docs update separately adds Azure Key Vault signing for the Windows desktop app.

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

01 **Initiator field in policy context** NEW 90

Policy context now includes `initiator` with `initiator.kind` and `initiator.id`, letting boundary rules distinguish scheduled routine runs from human sessions — e.g. `deny: initiator.kind == "routine" && intent == "run_command"`. `handoff` is its own initiator kind reserved for Bot-to-Bot delegation.

Block a Bot from running shell commands during unattended scheduled runs while still allowing the same Bot to run them when a human is watching.

```text
deny: initiator.kind == "routine" && intent == "run_command"
```

— Named config fields plus a runnable policy rule example v0.0.11

02 **Credential content inspection for Basic/Bearer tokens**

IMPROVED 70

Tool arguments starting with Basic or Bearer are now refused only when the value is shaped like a real credential: base64 decoding to `user:password` after Basic, or a token of at least 16 characters containing a digit, punctuation, or mixed case after Bearer.

— Detection logic detailed but nothing for reader to run v0.0.11

03 **Dry-run labeling and refusal recording on Audit page** IMPROVED 70

The Audit page now shows the 'dry-run: recorded, not enforced' label only on rows the policy refused and dry-run let through, and records tool calls refused for credential material with `carriedOut: false` in every mode.

— Names exact field but only points to a UI page v0.0.11

THINNER COVERAGE BELOW

04 **Worker check-in status on Routines page** IMPROVED 55

The Routines page now reports whether a worker process has ever checked in and when the last sweep occurred, instead of showing a waiting routine with a next-run time when no worker is running.

— Clear UI behavior change, no named config surface v0.0.11

05 **Azure Key Vault signing for Windows app** NEW 55

OpenBot adds Azure Key Vault-backed signing and verification for the Windows desktop app and NSIS installer, documented in `windows-signing.md`.

— Names doc file but no setup steps given product docs

06 **Refusal reason forwarding in Python LangGraph Bot** IMPROVED 45

The Python LangGraph Bot now forwards the deployment's refusal reason to its model, matching the existing behavior of the TypeScript LangGraph Bot.

— Brief parity note with no mechanism detail v0.0.11

WAS THIS USEFUL? 👍 👎

assistant-ui

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗ RANK

Assistant-ui provides React components and state management for chat interfaces with streaming, tool calls, and attachments.

> assistant-ui 0.15.20 focuses on internal performance and metadata: caching for external message conversion, automatic task derivation from nested tool-call conversations, and modality tagging for voice messages.

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

01 **Caching for external message conversion** NEW 85

`unstable_createExternalMessageConversionCache`, exported from `@assistant-ui/react` and `@assistant-ui/react-native`, lets callers opt into caching for `convertExternalMessages` so an unchanged source message reuses its existing `ThreadMessage` object instead of being rebuilt. It is used by `react-langchain` to stop streamed tokens from rebuilding every message in a nested subagent transcript.

— Names exact exported function, packages, and consuming use case.

`@assistant-ui/react@0.15.20`

THINNER COVERAGE BELOW

02 **Automatic task derivation from tool calls** NEW 45

`thread.tasks` is now derived automatically from tool calls that carry nested conversations, scoped to a task scope.

— Names the field but no usage or scope mechanism detail. `@assistant-ui/react@0.15.20`

03 **Voice modality metadata on messages** NEW 42

Voice transcript messages are now marked with `metadata.modality`, allowing callers to distinguish voice-originated messages from text ones.

— Names the metadata field but no example of use. `@assistant-ui/react@0.15.20`

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

DMLC XGBoost

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗ RANK

Scalable, Portable and Distributed Gradient Boosting (GBDT, GBRT or GBM) Library, for Python, R, Java, Scala, C and more. Runs on single machine, Hadoop, Spark, Dask, Flink and DataFlow

> XGBoost added a new probabilistic regression objective for estimating conditional mean and variance, and extended Python compatibility to 3.11.

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

01 **Normal-distribution regression objective** NEW 75

New `reg:normal` objective performs conditional normal distribution regression, outputting both mean and log-variance predictions. It pairs with a new `normal-nloglik` evaluation metric, which serves as the default metric for this objective.

Train a model that estimates both the conditional mean and uncertainty (log variance) of a scalar response — useful for probabilistic regression tasks.

python

```python
import xgboost as xgb

params = {
    'objective': 'reg:normal',
    'eval_metric': 'normal-nloglik',
    'n_estimators': 100,
}

model = xgb.XGBRegressor(**params)
model.fit(X_train, y_train)

# predictions shape: (n_rows, 2) -> [mean, log_variance]
preds = model.predict(X_test)
```

— Names objective, metric, and shows runnable example product docs

THINNER COVERAGE BELOW

02 **Python 3.11 support** NEW 35

XGBoost v3.4.2 adds support for Python 3.11.

— Bare compatibility note with no further detail v3.4.2

WAS THIS USEFUL?  

BerriAI LiteLLM

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗

RANK

LiteLLM is an AI gateway that calls over 100 model APIs in OpenAI format with cost tracking, guardrails, load balancing, and logging.

> LiteLLM v1.101.0 adds OIDC workload identity federation for OpenAI, native GigaChat API passthrough, a Responses API token-counting endpoint, and a round of upgrades to the complexity auto-router and shadow-eval system.

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

01 **Token counting endpoint for Responses API** NEW 80

Adds `/v1/responses/input_tokens` endpoint for token counting on Responses API requests, callable as `POST /v1/responses/input_tokens` with a model and input payload to count tokens before running a full inference call.

Count input tokens for a Responses API payload before committing to a full inference call.

```
$ curl -X POST http://localhost:4000/v1/responses/input_tokens \
  -H 'Authorization: Bearer sk-1234' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gpt-4o", "input": "How many tokens is this prompt?"}'
```

— Named endpoint with a runnable curl example v1.101.0

02 **Webhook-only budget alerting** NEW 75

Adds `ALERTING_WEBHOOK_URL` environment variable so budget alerts fire on webhook-only alerting configurations, set via e.g. `export ALERTING_WEBHOOK_URL=https://hooks.example.com/litellm-budget-alerts`.

Receive budget-exhaustion webhook alerts without configuring any other alerting channel.

```
$ export ALERTING_WEBHOOK_URL=https://hooks.example.com/litellm-budget-alerts
```

— Named env var with a runnable example v1.101.0

03 **Complexity router escalation and modality routing** NEW

70

The complexity auto-router gains `classification_mode` config to skip the classifier on continuation turns, context-window escalation that automatically promotes oversized prompts to a tier with sufficient context before dispatch, and opt-in modality-based capability routing that directs image requests to vision-capable model tiers. The UI adds controls for context-window escalation and modality routing on the auto-router create and edit forms, plus a unified classification-frequency picker for complexity auto-routers.

— Names config key and mechanisms but only a UI path, no exact flags v1.101.0

THINNER COVERAGE BELOW

04 **Budget window usage on key info endpoint** IMPROVED 50

Adds budget window usage display on `GET /key/info`.

— Named endpoint but minimal detail on what's shown v1.101.0

05 **Argo CD hooks in Helm chart** NEW 50

Adds an Argo CD `PreSync` hook and rollout strategy knobs to the componentized Helm chart.

— Names the hook but not the specific chart values v1.101.0

06 **Shadow-eval targeting and multi-router comparison** NEW

45

Adds shadow-eval targeting for teams and users, enabling JWT-auth traffic to be evaluated, and shadow-eval job support for comparing multiple auto-routers on sampled traffic from a single job.

— Explains scope but no config keys or endpoints given v1.101.0

07 **GigaChat API passthrough routes** NEW 40

Adds native GigaChat API passthrough routes with spend logging.

— Thin description, no route paths or config named v1.101.0

08 **OIDC workload identity federation for OpenAI** NEW 40

Adds workload identity federation (OIDC token exchange) support for OpenAI calls.

— States the mechanism name but no config or setup detail v1.101.0

09 **GLM-5.3 pricing for Friendli** NEW 40

Adds `zai-org/GLM-5.3` and `zai-org/GLM-5.3-Flash` model pricing for Friendli.

— Names exact models but no pricing figures given v1.101.0

10 **Router metadata in spend logs** IMPROVED 25

Persists router metadata in spend logs for internal router models.

— Very thin, no fields or schema named `v1.101.0`

WAS THIS USEFUL?  

Eigen Labs Darkbloom

Feedback ⓘ 1 RELEASE · 2026-09-15

NOTES ↗ CODE ↗ RANK

Private Inference Network on Idle Macs

> Darkbloom's v0.9.3 release turns on SSD prefix caching by default for GPT-OSS 20B and adds a suite of cache-routing refinements, alongside a new App Attest shadow archive for tracking machine identities.

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

01 **Prefix cache write priority limiting** IMPROVED 85

Limits first-seen checkpoint writes to a 90% share of the write budget, reserving capacity for prefixes seen again within the cache TTL; when the novel-share is exhausted this now reports `write_priority_limited`, distinct from `write_rate_limited`.

— Concrete percentage and two named status codes readers can check v0.9.3

02 **App Attest shadow archive with stable machine identities** NEW 75

Adds an App Attest shadow archive that assigns stable server-assigned machine identities, backfills historical macOS adoption inventory, and exposes an authenticated admin dashboard with complete-record downloads — while keeping APNs and MDM authoritative for routing, trust, and payments.

— Names components and scope but no dashboard path or API given v0.9.3

03 **SSD prefix caching enabled by default for GPT-OSS 20B**

IMPROVED 70

Darkbloom now enables SSD prefix caching by default for `gpt-oss-20b`, so repeated prompts and changed-question branches automatically reuse full-attention and sliding-window checkpoints without any opt-in.

— Explains checkpoint reuse mechanism but no config to toggle it v0.9.3

04 **Repeated-prefix routing preference (prefix_affinity)** NEW 65

Adds a repeated-prefix routing preference, `prefix_affinity`, so the coordinator steers repeated prefixes to a stable cache-capable provider among otherwise equivalent cost, queue, and pending-work candidates.

— Names the flag and mechanism but no config file shown v0.9.3

05 **Cache opportunity diagnostics** NEW 60

Adds cache opportunity diagnostics that report per-model reasons and counts for repeated-prefix demand, usable holders, and routing selection, distinguishing routing opportunities from actual cache hits.

— Describes metrics reported but no endpoint or command to view them v0.9.3

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗

RANK

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

> OpenAI Python SDK v3.14.0 normalizes errors raised while reading streams for more consistent exception handling.

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

01 **Normalized stream reading errors** IMPROVED 31

Errors raised while reading streams are now normalized so stream consumers receive consistent exception types regardless of where the error originates.

— Describes the change but no specific exception types or code paths named. v3.14.0

WAS THIS USEFUL? 👍 👎

fal

Feedback ⓘ 2 RELEASES · 2026-09-09 → 2026-09-14 NOTES ↗ RANK

fal provides hosted generative-media model APIs and a serverless GPU platform for deploying custom inference workloads.

> fal added billing metadata for WebSocket endpoints and a way to interactively test any endpoint of a deployed serverless app from the dashboard.

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

01 **Billing headers for WebSocket endpoints** NEW 85

WebSocket endpoints can now declare billing units via the `x-fal-billable-units` header on the 101 upgrade response, or signal deferred reporting with `x-fal-billable-units-webhook` (set to `1`) on the same response. A new `POST /requests/billable-units/{request_id}` endpoint reports total billable units after a WebSocket session concludes.

— Names exact headers and endpoint but no usage example

Billing Headers for WebSocket Endpoints

02 **Endpoint testing in Dashboard Playground** NEW 65

A new `Testing > Playground` section on each deployed serverless app's dashboard lets you select an available endpoint of that app, fill in its generated input form, and run a request using existing Playground authentication and billing behavior, without leaving the app view.

Test any endpoint of a multi-endpoint serverless app interactively without writing a client or leaving the dashboard.

📍 In the app dashboard, go to Testing › Playground, select an available endpoint from the dropdown, fill in the generated input form, and click Run.

— Clear UI navigation path with usage steps included

Switch Between App Endpoints in the Das…

WAS THIS USEFUL? 👍 👎

OpenRouter

Feedback ⓘ 1 RELEASE · 2026-09-10 NOTES ↗ RANK

OpenRouter is a routing gateway that exposes hundreds of models from many providers behind one OpenAI-compatible API with failover and unified billing.

> OpenRouter introduces Presets, a config-as-code construct for bundling and versioning LLM call configuration that can be referenced by slug across apps.

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

01 **Presets for versioned LLM call configs** NEW 80

Presets bundle models, system prompts, provider routing, and sampling parameters into a named, versioned unit referenceable as `@preset/your-slug` in any API call, e.g. by setting `"model": "@preset/your-slug"` in a chat completions request. Updating a Preset in the dashboard propagates the change to every app that references it without requiring a redeploy.

Reference a saved Preset in a chat completions call to apply your versioned model, routing, and sampling config without hardcoding parameters.

```
$ curl https://openrouter.ai/api/v1/chat/completions \
  -H 'Authorization: Bearer <your-api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "@preset/your-slug", "messages": [{"role":
"user", "content": "Hello"}]}'
```

— Names exact syntax and mechanism, includes runnable example snapshot-20260915

WAS THIS USEFUL? 👍 👎

emisar

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗ RANK

Emisar lets AI agents securely run infrastructure actions through MCP, with approval controls that protect production systems.

> Emisar v0.49.0 adds a policy-gated `review` object with per-vote audit trails to MCP run responses and the run page, enforces `set -e` across every packaged script, and realigns risk tiers across packs (retiring older `fail2ban` and `cockroach` pack versions) alongside smarter diagnostic pack discovery and JSON pack error handling.

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

01 Per-vote approval audit trail in MCP and run page NEW

85

Adds a `review` object to `wait_for_run` and `recent_runs` MCP responses for policy-gated runs, exposing dispatch rationale, the trusted command line, votes, and the override receipt under the `view_runs` permission; the object is bounded to 64 KiB in JSON-encoded bytes after masking so a justification quoting a secret cannot push a `wait_for_run` page past its size budget. The run page now lists per-vote approval rows — reviewer names, notes, and the tally — repainting live as votes arrive, with an override receipt shown separately from an ordinary approval.

— Names exact MCP fields, permission, and size bound v0.49.0

02 Risk tier alignment across packs BREAKING

78

Unifies the risk tier for reloading a config to `high` in every pack, and aligns the cancel-query tier across Postgres and CockroachDB; `fail2ban` and `cockroach` pack versions below 0.2.0 that previously auto-ran those actions are retired. Also adds `member_invite` in the admin pack to the `high` risk tier, matching every other membership-change action.

— Names retired pack versions and specific tier changes v0.49.0

03 Diagnostic pack discovery via symptom keywords IMPROVED

70

The six on-host diagnostic packs (`linux-core` , `debugging` , `systemd-deep` , `process-forensics` , `fs-search` , `network-tls`) are enriched with operator-typed symptom keywords so searches reach them first; `podman` , `zfs` , `wireguard` , and `nodejs-pm2` are suggested on hosts that run them.

— Lists all six packs and four suggested tools v0.49.0

04 `set -e` enforcement across pack scripts IMPROVED

65

Enforces `set -e` (`set -euo pipefail` under bash) on every packaged script, with the packs gate linting for it, so a failed step ends the action instead of silently exiting 0.

— Explains mechanism and enforcement point clearly v0.49.0

THINNER COVERAGE BELOW

05 **`--fail-with-body` flag for JSON packs** IMPROVED 55

JSON packs now return the provider error body via `--fail-with-body` instead of discarding it.

— Names exact flag and behavior change v0.49.0

06 **Worked examples for Stripe and Braintree actions** IMPROVED 35

Every Stripe and Braintree action ships a worked example.

— Thin description with no further specifics v0.49.0

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

vMLX

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗ RANK

The vMLX server runs compressed MLX models on Apple Silicon with disk caching, paged memory, continuous batching, and hybrid SSM scheduling.

> vMLX 1.6.59 adds SSD cache management directly in the Chat view, refines streamed reasoning and context-budget error handling, improves Qwen vision media positioning, and upgrades its desktop runtime.

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

01 **SSD cache controls in Chat view** NEW 57

Surfaces SSD capacity notices and a Clear SSD Cache control directly in the active Chat view, letting users manage on-disk model cache without leaving the chat interface.

— Names the exact UI control but not its full mechanism v1.6.59

02 **Qwen vision media position derivation** IMPROVED 45

Derives Qwen media positions from complete vision blocks, with numerical and sparse-prefill experiments remaining opt-in.

— Names Qwen and sparse-prefill but gives no flag to enable it v1.6.59

03 **Streamed reasoning and context budget errors** IMPROVED 40

Preserves streamed Responses reasoning around tool-call buffering, and context errors now distinguish between input and output token budgets.

— Describes behavior change but no config or command to invoke it v1.6.59

04 **Desktop runtime upgrade** IMPROVED 35

Upgrades the desktop runtime to Electron 44.3.0 and SQLite 13.0.3.

— Exact version numbers given but no user-facing action v1.6.59

WAS THIS USEFUL? 👍 👎

mlxtop

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗ RANK

The mlxtop terminal UI monitors local LLMs on Apple Silicon, showing loaded models, memory use, GPU activity, and generation speed.

> mlxtop's latest updates center on a new client-side usage-file protocol for exact token and latency tracking, three new Overview time-series charts (process memory, queue depth, and per-request prompt load), and expanded Linux platform support alongside a redesigned three-view TUI.

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

01 Usage-file protocol for exact token tracking NEW 95

The `MLXTOP_USAGE_FILE` environment variable ingests a client-written JSONL file of per-completion token counts, enabling exact prompt/output tracking for MLX-LM, Ollama, LM Studio, LocalAI, llama-server, and KoboldCpp without any provider-side changes. Each record can carry `prompt_tokens_details.cached_tokens` to surface per-request cache reuse; v1.1.0 documents this as a counters-only usage file for client-reported responses.

Track exact per-completion token counts for an MLX-LM server without any provider-side changes — write a usage record from your client after each completion, then point mlxtop at the file.

```
$ MLXTOP_USAGE_FILE=/absolute/path/usage.jsonl
./target/release/mlxtop
```

— Names exact env var, file format, six providers, and JSON field v1.1.0

02 Prompt load panel in Overview NEW 80

A compact prompt-load panel shows latest prompt size, delta from the previous request, and cache-reuse coloring, with a bar-chart history of up to 240 requests browsable via `↑` / `↓`, PgUp/PgDn, Home, End, and `r` to reset. It also reports exact selected counts, freshness indicators, growth comparisons, and cached/uncached segment breakdown when reported.

— Named keyboard controls make it directly usable v1.1.0

03 Explicit provider selection via environment variables NEW 75

`MLXTOP_PROVIDER` and `MLXTOP_PROVIDER_PORT` explicitly select and configure a local inference server — e.g. KoboldCpp on port 5001 or llama-server on port 8080 — instead of relying on auto-detection.

Pin mlxtop to a specific llama-server instance running on a non-default port instead of relying on process auto-detection.

```
$ MLXTOP_PROVIDER=llama.cpp MLXTOP_PROVIDER_PORT=8081
./target/release/mlxtop
```

Force mlxtop to monitor a llama.cpp server running on a non-default port instead of auto-detecting the provider.


```
$ MLXTOP_PROVIDER=llama.cpp MLXTOP_PROVIDER_PORT=8081
./target/release/mlxtop
```


— Exact env vars and runnable command given `product docs`

04 **First-token latency chart** NEW 75

A time-series chart in Overview is driven by `timings.time_to_first_token_ms` from the usage JSONL envelope, using a fixed 0–30-second scale with overflow markers; gaps appear where timings are missing. v1.1.0 describes this as an optional display shown only when a client supplies an explicit timing.

Supply first-token latency data so the latency chart populates — append this envelope to the usage JSONL file after each completed request.

```json
{"provider":"mlx-lm","request_id":"req-123","model":"local-
model","observed_at":1700000000,"usage":
{"prompt_tokens":32768,"completion_tokens":120,"prompt_tokens_det
ails":{"cached_tokens":24576}},"timings":
{"time_to_first_token_ms":1250}}
```

Send first-token latency data to mlxtop so the time-to-first-token chart populates in Overview.

```json
{"provider": "llama.cpp", "request_id": "req-001", "observed_at":
"2024-06-01T12:00:00Z", "usage": {}, "timings":
{"time_to_first_token_ms": 1250}}
```

— Mechanism, scale, and data source specified, plus example payload `v1.1.0`

05 **macOS process memory tracking** NEW 75

A process memory time-series chart plots the LLM process's OS physical footprint against total system RAM (with current PID in the header) on wide terminals. Underlying telemetry via `proc_pid_rusage` surfaces footprint, lifetime peak, and signed growth (bytes/sec) for the process with the largest RSS, visible on the throughput card and static report, requiring no changes or restarts to the serving runtime.

— Detailed mechanism and metrics but UI-only, no command `v1.1.0`

---

06 **Linux platform support for GPU and thermals** NEW 75

Adds Linux GPU support via `nvidia-smi` (device name, utilization, VRAM used/total, temperature) and thermals via `/sys/class/thermal`, plus Linux source build support drawing from `/proc`, `/sys`, and optional `nvidia-smi` metrics.

— Names all Linux data sources but no exact install command `v1.1.0`

---

07 **llama-server metrics and per-slot ingestion** NEW 75

Reads llama-server aggregate rates from `/metrics` when the server is started with `--metrics`, and per-slot data from `/slots`.

— Names exact endpoints and required startup flag `product docs`

---

08 **Three-view TUI layout with keyboard navigation** NEW 75

A three-view layout — Overview (`1` / `o`), MLX Top (`2` / `t`), and Journal (`3` / `j`) — adds per-view keyboard controls including sort cycling (`s`), text filter (`f` / `/`), and journal event filters (`f` / `]` / `[`).

— Named views and exact keybindings for navigation `product docs`

---

09 **Queue depth chart in Overview** NEW 65

A time-series chart plots active requests (cyan) and waiting requests (yellow) on a shared fixed 0–16 scale, with overflow markers and `=` for overlapping values, preserving gaps when telemetry is unavailable or stale.

— Full visual mechanism described but purely display, no controls `v1.1.0`

---

10 **Structured log paths and custom log location** NEW 65

Logs write to `~/Library/Logs/mlxtop/mlxtop.log` on macOS and `~/.local/state/mlxtop/mlxtop.log` on Linux (respecting `$XDG_STATE_HOME`); the `MLXTOP_LOG_PATH` environment variable redirects crash-diagnostic logs to a custom path for centralized collection.

Redirect crash-diagnostic logs to a custom path for centralized collection.

```
$ MLXTOP_LOG_PATH=/var/log/mlxtop/session.log
./target/release/mlxtop
```

— Exact default paths and override variable with example product docs

THINNER COVERAGE BELOW

11 **One-shot snapshot mode** NEW 55

The `mlxtop --once` flag captures a one-shot LLM serving snapshot, useful for running over SSH without keeping an interactive session open.

Capture a one-shot LLM serving snapshot over SSH without keeping an interactive session open.

```
$ mlxtop --once
```

— Exact runnable flag but no further description v1.1.0

12 **Remote endpoint authorization override** NEW 40

`MLXTOP_ALLOW_REMOTE_AUTH=1` permits connecting to an intentionally remote oMLX endpoint.

— Only the env var name is given, no further mechanism product docs

WAS THIS USEFUL? 👍 👎

OpenWhispr

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ RANK

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

> OpenWhispr 1.10.1 adds quick-start note creation from anywhere in the app, a built-in emoji picker, and broader video/audio upload support, alongside reworked team/group management and a shorter Detailed Notes prompt.

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

01 **Video and audio upload format support** IMPROVED 75

Extends upload to accept MPEG audio, `.mp4`, `.mov`, `.mkv`, and `.3gp` files with a speech track; rejected drops now display a reason so users can identify unsupported formats.

Upload a video recording of a meeting for transcription when you forgot to use the in-app recorder.

📍 In the app, drag an `.mp4`, `.mov`, `.mkv`, or `.3gp` file onto the upload target; if the file is rejected, read the inline reason to identify the unsupported format.

— Exact formats named plus a runnable drag-and-drop example. v1.10.1

02 **Quick-start note and recording button** NEW 70

Adds a split `+ New note` button that creates a note and starts recording from any view, with a chevron revealing Assistant chat and Team space options; the tray menu and dictation pill gain the same quick actions (tray: Start listening, Start meeting recording, Ask assistant).

Start a note and recording instantly from the tray without switching to the main window.

📍 Click the tray icon › Start listening to begin dictation, or Start meeting recording to capture a meeting; use Ask assistant to open the assistant chat directly from the tray.

— Names exact UI/tray actions and shows a usage example. v1.10.1

03 **Teams renamed to Groups with unified member roster** IMPROVED

70

Renames Teams to Groups across workspace copy in all 11 locales, and consolidates a space's member list into a single flat roster (including group-granted access, marked via Design) under one Settings dialog; creating a space no longer requires a group.

— Names the rename scope, dialog, and behavior change precisely. v1.10.1

04 **Detailed Notes prompt rewrite** IMPROVED

65

Rewrites the Detailed Notes prompt with a shorter format tested against 30-, 60-, and 90-minute meetings; action items now use a `- [ ] Action — Owner` shape compatible with the editor's owner-tagging feature.

— Concrete format and testing scope given but no user-facing steps. v1.10.1

05 **Built-in emoji picker** NEW

60

Adds a built-in emoji picker that is searchable, categorised, keyboard-navigable, and shows recents; it is translated into ten languages and replaces the OS panel, which never worked on Linux.

— Mechanism described well but no command or exact navigation given. v1.10.1

THINNER COVERAGE BELOW

06 **AI Summary tab default and renamed raw tab** IMPROVED

50

Notes with an existing summary now open on the AI Summary tab (formerly Enhanced) instead of the raw tab; the raw tab is renamed Your notes.

— Names the tabs and rename but is a small UI change. v1.10.1

07 **Invite your team sidebar entry** NEW

40

Adds an Invite your team row in the sidebar for workspace owners and admins, opening the invitation dialog.

— Named UI element with clear navigation but minimal detail. v1.10.1

WAS THIS USEFUL? 👍 👎

`llama.cpp` Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ CODE ↗

RANK

Llama CPP is a C/C++ library and set of command-line tools that runs large language models locally.

> llama.cpp v0.4.1 adds support for three new model architectures, introduces structured JSONL logging and a Q/K/V-fusing conversion flag, and bumps ggml to v0.24.0 with a new precision-control API and expanded backend coverage, while removing the deprecated `--mmap` / `--mlock` / `--direct-io` flags in favor of `--load-mode`.

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

01 QKV tensor fusion during GGUF conversion NEW 85

Adds a `--fuse-qkv` conversion flag that fuses Q/K/V tensors into a single QKV tensor when converting a HuggingFace checkpoint to GGUF, for inference efficiency.

Convert a HuggingFace checkpoint to GGUF with Q/K/V attention tensors fused into a single QKV tensor for inference efficiency.

```
$ python convert_hf_to_gguf.py --fuse-qkv /path/to/hf-model --outfile model-fused.gguf
```

— Names flag and mechanism, includes runnable conversion command v0.4.1

02 Structured JSONL logging for the server NEW 80

Adds a `--log-jsonl` flag and a `LOG_JSON` macro to emit structured JSONL logs from llama-server, intended for machine-readable log pipelines such as SIEM ingestion.

Capture structured JSONL logs from llama-server for ingestion into a SIEM or log aggregator.

```
$ llama-server -m model.gguf --log-jsonl 2>>server-logs.jsonl
```

— Names exact flag and shows runnable capture command v0.4.1

03 **Removal of legacy memory-mapping flags** BREAKING 80

Removes the deprecated `--mmap`, `--mlock`, and `--direct-io` arguments; `--load-mode` is now the sole interface for controlling memory-mapping behaviour.

— Names all removed flags and their replacement clearly v0.4.1

04 **Sampler chain API return-type change** BREAKING 65

Changes llama_sampler_chain_n()'s return type from `int` to `int32_t`, which may require callers to update variable declarations.

— Names exact function and migration impact, no example v0.4.1

THINNER COVERAGE BELOW

05 **New model architecture support: Maple, Hy 4, Kimi-K3** NEW 58

Adds Maple 20B-A1B ternary MoE architecture support for CPU inference, Tencent Hy 4 (`hy_v4`) preview architecture support, and Kimi-K3 recurrent-state rollback support.

— Names each model and architecture detail but no usage steps v0.4.1

06 **ggml v0.24.0 with precision-control API** IMPROVED 55

Bumps ggml from v0.23.0 to v0.24.0, bringing a new precision-control API, expanded Vulkan/SYCL/Hexagon/OpenCL backend coverage, and performance improvements across CPU, CUDA, and Metal.

— Names versions and backends but no usage instructions v0.4.1

07 **Default device selection for mmproj and draft models** IMPROVED 55

Makes mmproj and draft devices default to the global `--device` selection when not explicitly set with `-mmdev`.

— Names both flags, clear default behaviour change v0.4.1

08 **Router child-process monitoring helpers** NEW 40

Adds `server_subproc` and `waiter` to `server-common.h` for monitoring router child processes.

— Names exact symbols but no behaviour or usage detail v0.4.1

09 **Reasoning preservation enabled by default** IMPROVED

40

Enables `--reasoning-preserve` by default in the server.

— Names flag but gives no further mechanism v0.4.1

10 **Model downloads allowed past --models-max limit** IMPROVED

40

Allows model downloads when the `--models-max` limit is already reached.

— Names the config flag but no further detail v0.4.1

11 **Faster chat message rendering** IMPROVED

40

Improves chat message rendering performance with lazy mounting and compositor-friendly animations.

— Describes technique but no named surface or metric v0.4.1

└──▷ **BREAKING ON UPGRADE**

! The `--mmap`, `--mlock`, and `--direct-io` args are removed; use `--load-mode` instead.

! llama_sampler_chain_n() now returns `int32_t` instead of `int`, which may require callers to update variable declarations.

WAS THIS USEFUL? 👍 👎

KTransformers

Feedback ⓘ v0.7.1 NOTES ↗

KTransformers provides kernels for language-model inference and fine-tuning with heterogeneous CPU and GPU execution.

KTransformers v0.7.1 adds Qwen3-VL BF16 LoRA fine-tuning and native RAWINT4 Kimi K2.5/K2.6 LoRA fine-tuning via LLaMA-Factory.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.7.1 https://github.com/kvcache-
ai/ktransformers.git
# already have the repo? check out this version:
$ git checkout v0.7.1
```

Adds BF16 image-text LoRA fine-tuning for `Qwen3-VL-30B-A3B-Instruct` and `Qwen3.5-35B-A3B`, covering vision, language, and routed-expert modules via LLaMA-Factory integration.

Adds native RAWINT4 LoRA fine-tuning for Kimi K2.5 and K2.6 using original packed expert weights, eliminating the need for full-model BF16 expansion.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Pinecone

Feedback ⓘ 1 RELEASE · SEEN 2026-09-15 NOTES ↗ RANK

Pinecone is a managed vector database that stores and queries embeddings for AI applications.

> Pinecone's release window centered on new Nexus curation and query pipelines with a fresh embedding model, alongside data-plane expansion covering dedicated read nodes, metadata indexing controls, a schema-based Documents API, and fuller organization/service-account management APIs.

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

01 Organization and service-account management APIs NEW

86

Adds full organization, project and service-account management APIs, including a `rotate-secret` endpoint (`POST /organizations/<ORG_ID>/service-accounts/<SA_ID>/rotate-secret`) that revokes the previous OAuth client secret within seconds and returns the new secret only once.

Rotate a service account's OAuth client secret after a potential exposure — the previous secret is revoked within seconds and the new one is returned only once.

```
$ curl -X POST
'https://api.pinecone.io/organizations/<ORG_ID>/service-
accounts/<SA_ID>/rotate-secret' \
  -H 'Api-Key: <YOUR_API_KEY>'
```

— Exact endpoint and runnable secret-rotation command provided. snapshot-20260915

02 llama-text-embed-v2 embedding model NEW

60

Adds the `llama-text-embed-v2` embedding model, built on the Llama 3.2 1B architecture by NVIDIA Research, optimized for high retrieval quality with low-latency inference.

— Model and architecture named but no invocation example shown. product docs

THINNER COVERAGE BELOW

03 Pinecone Nexus curation and query pipelines NEW

50

Adds Pinecone Nexus curation, a write path that incrementally turns sources into chunks and artifacts driven by a manifest, and Pinecone Nexus queries, which gather evidence and compose grounded, cited answers.

— Conceptual description only, no command or config surface given. product docs

04 **Multitenancy design guidance** NEW 30

Publishes guidance for multitenancy design covering namespace-per-tenant isolation versus metadata filtering tradeoffs.

— Guidance only, no concrete config or command included. snapshot-20260915

└──▷ **BREAKING ON UPGRADE**

! API version `2026-07` introduces schema-based indexes via the Documents API; existing code must be reviewed for compatibility and migrated to the new schema.

WAS THIS USEFUL? 👍 👎

timgordontg engrim

Feedback ⓘ 1 RELEASE · 2026-09-15

NOTES ↗ RANK

The Universal Cross-Model Episodic Memory Standard. Local-first, project-scoped SQLite engine for Google Antigravity, Claude Code, Cursor, Codex CLI, and OpenCode.

> engrim v1.4.6 brings its four MCP tools into StructuredContent compliance by serving JSON payloads alongside text content.

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

01 **MCP StructuredContent compliance for tool calls** IMPROVED 55

Tool call results for `engrim_add`, `engrim_recall`, `engrim_context`, and `engrim_review` now serve JSON payloads under `structuredContent` alongside `content[].text`, satisfying MCP StructuredContent compliance.

— Names exact tools and field but no usage example v1.4.6

WAS THIS USEFUL? 👍 👎

Weaviate

Feedback ⓘ 1 RELEASE · 2026-09-14 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 v1.38.15 introduces node-level query admission control to keep individual cluster nodes stable under heavy query load.

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

01 **Node-level query admission control** NEW 28

Weaviate adds node-level query admission control to prevent overloaded nodes from accepting new queries, improving cluster stability under high concurrency.

— Describes mechanism but no config keys or flags given v1.38.15

WAS THIS USEFUL? 👍 👎

Neo4j Agent Memory

Feedback ⓘ 1 RELEASE · 2026-09-14 CODE ↗

RANK

Agent Memory is a graph-native memory system for AI agents that stores conversations, builds knowledge graphs, and tracks reasoning traces with Neo4j.

> Agent Memory's python SDK (v0.6.0) introduces a Neo4jMemoryStore for Strands agents, giving them cross-session long-term recall, graph-native entity/preference tools, and a search result cap, plus reliability fixes for concurrent extraction and event-loop handling.

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

01 **Neo4jMemoryStore for Strands agents** NEW 80

Adds `Neo4jMemoryStore` for Strands agents, enabling cross-session long-term recall via MemoryManager(stores=[...]) with long-term search and server-side extraction writes; requires `strands-agents>=1.52` via the `[strands]` extra.

Give a Strands agent cross-session long-term memory backed by Neo4j, with graph-native entity and preference tools registered automatically.

python

```python
from neo4j import GraphDatabase
from strands import Agent
from strands.memory import MemoryManager
from agent_memory.strands import Neo4jMemoryStore

store = Neo4jMemoryStore(
    name="my_memory",
    settings=settings,          # owns and manages the client
    user_id="alice",
    max_search_results=20,
)

agent = Agent(
    tools=store.get_tools(),    # registers
my_memory_get_entity_graph
                                # and
my_memory_get_user_preferences
    memory_manager=MemoryManager(stores=[store]),
)
agent("What frameworks do I prefer for Python projects?")
```

— Names extra, version requirement and API, backed by a runnable example. python-v0.6.0

02 **Graph-native entity and preference tools via get_tools()** NEW

80

get_tools() on `Neo4jMemoryStore` registers `{name}_get_entity_graph` (multi-hop on Bolt, 1-hop on NAMS) and, when `user_id` is configured on Bolt, the user-scoped `{name}_get_user_preferences` tool.

Give a Strands agent cross-session long-term memory backed by Neo4j, with graph-native entity and preference tools registered automatically.

python

```python
from neo4j import GraphDatabase
from strands import Agent
from strands.memory import MemoryManager
from agent_memory.strands import Neo4jMemoryStore

store = Neo4jMemoryStore(
    name="my_memory",
    settings=settings,          # owns and manages the client
    user_id="alice",
    max_search_results=20,
)

agent = Agent(
    tools=store.get_tools(),    # registers
my_memory_get_entity_graph
                                # and
my_memory_get_user_preferences
    memory_manager=MemoryManager(stores=[store]),
)
agent("What frameworks do I prefer for Python projects?")
```

— Exact tool names and backend behavior given, shown in example usage. python-v0.6.0

03 **Reliability fixes for Strands integration** IMPROVED

70

Guards against double extraction when a `Neo4jMemoryStore` and a `Neo4jSessionManager` are paired in the same agent run. The owned client is also rebound automatically when Strands changes the event loop during synchronous Agent(...) runs, and a client passed via `client=` that hits a loop change now raises a named error instead of an opaque driver `RuntimeError`.

— Describes mechanism and before/after but no direct usage steps. python-v0.6.0

THINNER COVERAGE BELOW

04 **Search result cap on Neo4jMemoryStore.search()** NEW

`max_search_results` on Neo4jMemoryStore.search() caps the total rows returned across entities and preferences per call.

Give a Strands agent cross-session long-term memory backed by Neo4j, with graph-native entity and preference tools registered automatically.

python

```
from neo4j import GraphDatabase
from strands import Agent
from strands.memory import MemoryManager
from agent_memory.strands import Neo4jMemoryStore

store = Neo4jMemoryStore(
    name="my_memory",
    settings=settings,          # owns and manages the client
    user_id="alice",
    max_search_results=20,
)

agent = Agent(
    tools=store.get_tools(),    # registers
my_memory_get_entity_graph
                                # and
my_memory_get_user_preferences
    memory_manager=MemoryManager(stores=[store]),
)
agent("What frameworks do I prefer for Python projects?")
```

— Named parameter with clear purpose, but limited elaboration. python-v0.6.0

WAS THIS USEFUL? 👍 👎

okf-memory OKF Agent Memory

Feedback ⓘ

1 RELEASE · 2026-09-15 NOTES ↗ RANK

Git-native persistent memory for AI coding agents. 2 with sub-300µs in-memory BM25 search, embedded MCP server, and progressive disclosure.

> OKF Agent Memory v0.3.0 introduces the Agent Action Grammar for deterministic AI instruction files, a linter that enforces it, a dual-memory architecture splitting working and persistent knowledge, and a set of CLI commands to scaffold, symlink and validate the whole agent stack.

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

01 **AAG linting via okf agents lint** NEW 92

Adds the `okf agents lint` command enforcing five Agent Action Grammar rules: `AAG-001` (ASCII-only), `AAG-002` (RFC 2119 modal verbs), `AAG-003` (tool signature integrity), `AAG-004` (Mermaid diagrams), and `AAG-005` (400-token working-memory budget).

Lint an agent instruction file before committing it to catch token-budget violations and malformed tool signatures.

```
$ okf agents lint
```

— Names all five rule codes and a runnable command. v0.3.0

02 **Agent scaffolding, SSoT linking and stack validation** NEW 90

Adds `okf agents init --domain=...` to scaffold curated domain codices (`software`, `books`, `coaching`); `okf agents link` to create SSoT symlinks from a single `AGENTS.md` to `CLAUDE.md`, `.cursorrules`, `.windsurfrules`, and `.github/copilot-instructions.md`, with `--check` to audit existing symlinks and `--force` to repair them; and extends `okf validate` with an `--agents` flag to validate the knowledge graph conformance plus agent governance in one invocation.

Bootstrap a coaching project with a curated domain codex and immediately wire its AGENTS.md into every supported IDE with a single SSoT symlink pass.

```
$ okf agents init --domain=coaching --name="Executive Coaching Practice" && okf agents link
```

Run a full conformance check on both the knowledge graph and agent governance rules in one step, failing fast in CI if either layer is out of spec.

```
$ okf validate --agents --strict .
```

— Names every flag, target file and supported domain value with runnable commands. v0.3.0

03 **Dual-Memory Agent Architecture (DMAA)** NEW 65

Introduces a two-layer memory model: Layer 1 is a compact `AGENTS.md` working memory capped at 400 tokens, and Layer 2 is a persistent `knowledge/` domain memory queried on demand via BM25 or MCP tools.

— Concrete token cap and folder name but no direct command. v0.3.0

THINNER COVERAGE BELOW

04 **Agent Action Grammar (AAG) language** NEW 50

Introduces AAG, an EBNF-based pseudocode language for writing deterministic AI agent instruction files (`AGENTS.md` , `CLAUDE.md` , `.cursorrules`), claiming up to 70% token savings over natural-language rules.

— Describes the language and a figure but no command to try it. v0.3.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ 1 RELEASE · 2026-08-20

NOTES ↗ RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's latest snapshot adds CLI-driven LLM-as-judge evaluator creation, extends annotation queues to accept conversation threads alongside runs, and introduces pre-save testing for multi-turn evaluators, while deprecating the legacy feedback formula endpoints and removing legacy dataset comparison helpers from the SDKs.

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

01 Annotation-queue item endpoint supports RUN and THREAD items

NEW 88

`POST /annotation-queues/<id>/items` adds RUN items to an annotation queue, with the server resolving runs via ClickHouse or SmithDB and returning a standardized items envelope. The same endpoint now accepts `item_type THREAD` (with `thread_id` and `session_id`) to add conversation threads, and supports mixed batches of RUN and THREAD items in a single call.

Add a batch of run IDs to an annotation queue in one API call for downstream human review.

```
$ curl -X POST 'https://api.smith.langchain.com/annotation-queues/<queue_id>/items' \
  -H 'Content-Type: application/json' \
  -H 'x-api-key: <LANGSMITH_API_KEY>' \
  -d '{"items": [{"run_id": "<run_id_1>"}, {"run_id": "<run_id_2>"}]}'
```

Queue conversation threads for human review alongside regular runs by using item_type THREAD in the same batch.

```
$ curl -X POST 'https://api.smith.langchain.com/annotation-queues/<queue_id>/items' \
  -H 'Content-Type: application/json' \
  -H 'x-api-key: <LANGSMITH_API_KEY>' \
  -d '{"items": [{"item_type": "THREAD", "thread_id": "<thread_id>", "session_id": "<session_id>"}, {"run_id": "<run_id>"}]}'
```

— Endpoint, params and runnable curl examples all provided snapshot-20260915

02 **In-place replace flag for code evaluator uploads** NEW

75

The `evaluator upload --replace` flag updates existing code evaluator rules in place, avoiding a delete-before-create window if the replacement upload fails.

— Flag named with clear mechanism, no example shown snapshot-20260915

03 **Legacy feedback formula endpoints deprecated** DEPRECATED

75

The legacy feedback formula endpoints `POST /feedback/formulas`, `GET /feedback/formulas`, `GET /feedback/formulas/{feedback_formula_id}`, `PUT /feedback/formulas/{feedback_formula_id}`, and `DELETE /feedback/formulas/{feedback_formula_id}` are deprecated and scheduled for removal on 2026-08-20; existing feedback formulas should be migrated to composite evaluators.

— All endpoints and removal date named, migration path noted snapshot-20260915

04 **CLI command for LLM-as-judge evaluator creation** NEW

70

The `langsmith evaluator create-llm` CLI command defines structured LLM-as-judge evaluator rules from a prompt, schema, and model config file, targeting a specific project or dataset.

— Command and inputs named, but no runnable example given snapshot-20260915

THINNER COVERAGE BELOW

05 **Granular permission for dataset downloads** NEW

55

Dataset download access is now split into a new `download datasets` permission, enforced in both the application and APIs; the download button is disabled in the UI for users lacking the permission.

— Permission named but no API/config detail given snapshot-20260915

06 **Splits column in experiment comparison view** NEW 50

An optional, reorderable Splits (latest) column can be added to the experiment comparison view, showing each example's current dataset split assignments as chips.

– UI feature described without config key or command snapshot-20260915

07 **PEP 604 union return type support in code evaluators** IMPROVED 45

Code evaluator upload now accepts Python entrypoints annotated with PEP 604 union return types, for example `-> dict | None`.

– Concrete example type given but limited mechanism detail snapshot-20260915

08 **Legacy dataset comparison helpers removed from SDKs** BREAKING 45

Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; existing HTTP routes continue to work only for LangSmith UI clients.

– No specific helper names or migration steps given snapshot-20260915

09 **User-defined monthly trace limits** NEW 35

LangSmith now enforces user-defined monthly trace limits scoped to individual projects and users.

– No mechanism for configuring the limits described snapshot-20260915

└──▷ **BREAKING ON UPGRADE**

! The legacy feedback formula endpoints (`POST/GET /feedback/formulas` , `GET/PUT/DELETE /feedback/formulas/{feedback_formula_id}`) are deprecated and scheduled for removal on 2026-08-20; migrate existing feedback formulas to composite evaluators.

! Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; existing HTTP routes continue to work only for LangSmith UI clients.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Phoenix adds a REST endpoint for retrieving dataset splits, making dataset partitioning inspectable programmatically.

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

01 **Dataset splits retrieval endpoint** NEW 70

Adds `GET /datasets/{dataset_identifier}/splits` to retrieve the splits associated with a dataset, allowing programmatic inspection of how a versioned dataset is partitioned.

Retrieve all splits for a dataset by identifier to programmatically inspect how a versioned dataset is partitioned.

```
$ curl -X GET 'http://localhost:6006/datasets/my-eval-dataset/splits' -H 'Accept: application/json'
```

— Names exact endpoint and runnable curl example, but no mechanism detail.

arize-phoenix-v20.12.0

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ 1 RELEASE · 2026-09-14 NOTES ↗ RANK

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

> Langfuse v4.36.0 adds an OpenAI Responses relay to its AI gateway and ships several tracing and filtering UI refinements, including a toggleable session timeline, waterfall trace tooltips, and unified column visibility controls.

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

01 **OpenAI Responses relay in AI gateway** NEW 50

The Langfuse AI gateway now relays authenticated OpenAI Responses API requests, extending the gateway's supported upstream APIs.

– Names the API and gateway but no endpoint or config details v4.36.0

02 **Toggleable session timeline** IMPROVED 45

Users can now toggle the session timeline on or off in the session UI.

– Clear UI location named but no further mechanism v4.36.0

03 **Waterfall breakdown in trace tooltips** NEW 40

Trace breakdown tooltips gain a waterfall view, giving span-level timing visibility within the trace UI.

– Describes the UI addition without exact navigation steps v4.36.0

04 **Inline suggestions for filter search** IMPROVED 35

Filter search scope dropdowns across filter UIs are replaced with inline suggestions.

– States the change but not which filters or workflows affected v4.36.0

05 **Unified column visibility popover** IMPROVED 35

Column visibility controls are unified into a single popover shared across all tables.

– Describes the consolidation without a specific UI path v4.36.0

06 **Configurable event propagation block sizes** NEW 35

The worker gains a new configuration surface making event propagation block sizes configurable.

– No exact config key or flag name given v4.36.0

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ 1 RELEASE · 2026-09-01 NOTES ↗ RANK

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

> Braintrust's biggest addition this window is Loop-powered investigation and automation (Patterns and Loop automations) for surfacing and acting on trace trends, alongside a wave of new `bt` CLI subcommands for projects, trace management, and templates, plus SCIM-based org provisioning, new coding-agent tracing integrations, and two breaking changes to trace-integration versions and `bt sync pull` defaults.

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

01 **Custom CA bundle support for bt CLI tracing** NEW 90

`BRAINTRUST_CUSTOM_CA_BUNDLE` env var added to `bt` CLI v0.19.3+ lets Rust SDK tracing and data uploads trust private or corporate certificate authorities; the variable takes PEM certificate contents directly, not a file path.

— Exact env var, version, and value format given snapshot-20260915

02 **Project management subcommands in bt CLI** NEW 85

`bt projects create`, `bt projects view`, and `bt projects delete` subcommands manage Braintrust projects from the CLI with JSON output for automation; `view --json` returns project details without opening a browser.

List all projects in JSON for scripted automation — useful in CI pipelines that need to look up a project ID before pushing results.

```
$ bt projects view --json
```

— Named subcommands plus a runnable example command snapshot-20260915

03 **Patterns: scheduled Loop-powered trace investigations** NEW 80

Patterns is a scheduled Loop-powered investigation that surfaces recurring problems and trends in traces — including failure modes and cost regressions — with supporting traces and suggested fixes. It supports GPT-6 Astra and GPT-5.6 model families and requires data plane v2.13+.

— Explains mechanism and models but no exact command or path snapshot-20260915

04 **Session tagging for coding-agent traces** NEW 80

`--tag` flag and `BRAINTRUST_TAGS` env var added to `bt` CLI v0.19.3+ attach session tags to traces from Claude Code, Codex, Google Antigravity, OpenCode, and pi.

— Flag and env var named, no example invocation snapshot-20260915

05 **Loop automations with scheduled runs** NEW 70

Loop automations are scheduled Loop runs with their own instruction, model, and write permissions that leave a read-only thread, with results optionally sent to a Slack channel or webhook. Requires data plane v2.12.0+.

— Mechanism and requirement given, no setup steps snapshot-20260915

06 **Observability template export and apply** NEW 70

`bt observability template` commands (bt v0.19.0+) export a project's configuration as a portable JSON template and apply it to another project.

— Named command and format, no exact syntax shown snapshot-20260915

07 **Asynchronous Brainstore log search engine** IMPROVED 70

Brainstore log search now uses an asynchronous execution engine that streams results as segments finish scanning, covering full-text, phrase, and multi-clause ANY_SPAN() searches including negated ones; requires data plane v2.13.0+.

— Named search syntax and mechanism, no usage steps snapshot-20260915

08 **bt sync pull default history window change** BREAKING 70

`bt sync pull` now includes all matching experiment and dataset history unless you pass `--window` or set `BT_SYNC_WINDOW`; previously only a default window was pulled.

— Names flag and env var and before/after behavior snapshot-20260915

09 **Dashboard sections for organizing charts** NEW 65

Dashboard sections are named, reorderable, duplicatable, and collapsible chart groups that keep long dashboards readable; available on Pro and Enterprise plans and require data plane v2.13.0+.

— Plan and version requirements given, no navigation path snapshot-20260915

10 **CLI commands for managing trace integrations** NEW 60

`bt trace update` updates installed coding-agent tracing integrations without rewriting their tracing settings, and `bt trace doctor` warns about outdated plugins and mismatched package version ranges.

— Two named subcommands, thin description of each `snapshot-20260915`

11 **Blind human reviews project setting** NEW 60

The Blind human reviews project setting hides peer scores, comments, and aggregates for the entire review session; reviewers with the project Update permission are exempt.

— Behavior and exemption rule stated, no config path `snapshot-20260915`

12 **Domain auto-join and SCIM provisioning for org membership** NEW 60

Domain mapping automatically adds users from specific email domains to an organization on first sign-in, without requiring an invitation. SCIM provisioning (private preview) provisions organization members and optionally permission group membership from an identity provider based on SCIM group membership.

— Describes mechanism, no config keys or steps given `product docs`

THINNER COVERAGE BELOW

13 **Claude Code and Codex tracing integrations** NEW 55

Claude Code integration traces Claude Code sessions to Braintrust and accesses Braintrust data through the MCP server; the Codex integration does the same for Codex sessions.

— Names two integrations and MCP server, no setup detail `product docs`

14 **v2 tracing integrations for OpenCode and pi** BREAKING 50

`bt trace enable opencode` and `bt trace enable pi` now install v2 of their respective tracing integrations; see the v0.19.3 migration instructions before upgrading.

— Commands named but migration details not included `snapshot-20260915`

└──▷ **BREAKING ON UPGRADE**

! `bt trace enable opencode` and `bt trace enable pi` now install v2 of their respective tracing integrations — see the v0.19.3 migration instructions before upgrading.

! `bt sync pull` now includes all matching experiment and dataset history unless you pass `--window` or set `BT_SYNC_WINDOW`; previously only a default window was pulled.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik 2.2.62 improves throughput for bulk dataset and experiment uploads by defaulting to parallel processing.

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

01 **Multithreaded bulk dataset and experiment uploads** IMPROVED

35

Dataset and experiment bulk upload paths now default to 8 threads, increasing throughput for large-scale dataset and experiment operations.

— Names the default thread count but no config surface to adjust it 2.2.62

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

ai-safe2-framework

Feedback ⓘ 2 RELEASES · 2026-09-14

NOTES ↗ RANK

Ai-Safe2-Framework provides governance, risk, and compliance controls for securing agentic AI systems and non-human identities.

> AI SAFE² CLI shipped two new evidence-intake pipelines in this window: a portable agent system identity manifest with dual hashing and rich component/relationship/authority modeling, plus a provider-neutral harness evidence intake with seven-domain coverage assessment and SHA-256 source binding.

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

01 **Provider-neutral harness evidence intake** NEW 98

Adds `safe2 evidence harness SOURCE --output FILE` to validate and normalize a provider-neutral harness export from coding agents, automation harnesses, wrappers, or hooks, with a `--strict` flag that exits with status 1 when any coverage area is partial or missing while still writing the evidence file for inspection. It assesses a seven-domain coverage inventory (task lifecycle, tool calls, artifacts, tests, usage, environment state, completion claims), binds every accepted export to its exact source bytes via SHA-256 digest and size recording with a maximum source size of 1 MB, preserves producer, export identifier, source references, and collection basis as explicit attribution, and surfaces loss-aware recommendations identifying which evidence areas remain partial or missing rather than silently treating gaps as success. It establishes a stable provider-neutral integration seam for future adapters targeting Codex, Claude Code, Hermes, OpenClaw, and other agent harnesses.

— Command, flag, limits, and named future integrations fully specified.

2026-09-14_CLI_0.3.0_Provider_Neutral_H…

02 **Agent system identity manifest and evidence binding** NEW 96

Adds `safe2 evidence system SOURCE --output FILE` to normalize an attributed agent-system identity declaration into a portable manifest, and `safe2 evidence manifest FILE --subject-id ID --output FILE` to bind that manifest into a wider hashed evidence run; both accept a `--strict` flag that exits with status 1 when identity gaps remain, enabling CI gating. The manifest uses dual hashing — `source.sha256` binds exact submitted bytes while `system_fingerprint_sha256` fingerprints normalized system composition, stable across JSON key order, list order, and formatting changes — and models eight component categories (models, harnesses, tools, skills, memory, environments, policies, evaluators), a relationship graph (use,

execution, policy, evaluation, memory, delegation), an authority inventory (read, write, execute, network, delegate, approve, admin, unknown), coverage status (complete, partial, missing, not-applicable) and evidence basis (observed, declared, unknown) distinctions, plus coverage gap recommendations for remaining version, attribution, inventory, and topology gaps.

— Full commands, flags, and schema fields named; minor gap on downstream consumption.

2026-09-14_CLI_0.4.0_Know_What_You_Actu…

03 **Schema export for system identity contracts** NEW 65

Adds `safe2 schema export system-identity-source-v1` to export the provider-neutral input contract for agent system identity, and `safe2 schema export system-identity-manifest-v1` to export the normalized output contract for the same feature.

— Exact runnable commands given but no detail on schema contents.

2026-09-14_CLI_0.4.0_Know_What_You_Actu…

WAS THIS USEFUL? 👍 👎

Docker Sandboxes

Feedback ⓘ 1 RELEASE · 2026-09-15 NOTES ↗

RANK

Docker Sandboxes runs AI agents in isolated Docker environments with controlled filesystem, network, and tool access.

> Docker Sandboxes v0.43.0 reworks skill-store sharing around a tri-state `--skills` flag and `skills.defaultMode` config key (replacing the old `shareSkills` key), adds hosted model fallback and MCP header/OAuth controls, and ships a long tail of CLI, networking, git-kit and diagnostics improvements.

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

01 Tri-state skill store sharing control

BREAKING

91

Sandboxes gain a `--skills=off|readonly|readwrite` flag on `sbx create` / `sbx run` (with `--no-share-skills` kept as a deprecated alias for `--skills=off`) and a `skills.defaultMode` config key, set via `sbx settings set skills.defaultMode readonly`, that sets the default shared-store mount mode for all future sandboxes. Breaking: the `shareSkills` key in `sbxenv.yaml` is replaced by `skills` (accepting `off|readonly|readwrite`), so existing `shareSkills` entries must be migrated.

Create a sandbox isolated from the shared skill store trust boundary — useful when running untrusted agents that should not read or be affected by shared skills.

```
$ sbx run --skills=off claude
```

Change the default mount mode so all future sandboxes are read-only against the shared skill store, without needing to pass `--skills` every time.

```
$ sbx settings set skills.defaultMode readonly
```

— Names exact flags, config key and required migration. v0.43.0

02 MCP connection and authorization controls

BREAKING

90

`sbx mcp add` gains a `--header` flag to send custom request headers to remote MCP servers, with values substituted from the local secret vault (e.g. `${{ secrets.mcp_token }}`). MCP authorization adds `--skip-ssrf-check` for private OAuth discovery with fallback to advertised common OIDC scopes, plus `--no-scope`

support for local OAuth registrations. Breaking: MCP OAuth client secrets are renamed from `mcp:<server>.client_secret` to `mcp:<server>:client_secret`; secrets stored under the old dotted name are no longer read and must be re-set with `sbx secret set mcp:<server>:client_secret`.

Connect an MCP server that requires a custom Authorization header, pulling the value from the local secret vault.

```
$ sbx mcp add https://mcp.example.com --header 'Authorization: Bearer ${{ secrets.mcp_token }}'
```

— Flags, header syntax and secret-key migration all named. v0.43.0

03 **Hosted model providers and overflow fallback** NEW 83

`sbx run --model` gains `--overflow-provider` / `--overflow-model` flags to pair a local model with a hosted fallback for requests that exceed the local model's context limit, and `sbx run --provider` now accepts hosted models.dev providers served through llmman.

Run a local model with a hosted overflow fallback so oversized requests are handled automatically instead of failing.

```
$ sbx run --model ollama/llama3 --overflow-provider openai --overflow-model gpt-4o
```

— Runnable command given; fallback trigger threshold unspecified. v0.43.0

04 **Environment file variable references for per-project mounts** NEW 81

Environment files gain `${{ env.projectDir }}` and `${{ env.fileDir }}` variable references, letting `~/.sbxenv.yaml` mount each project's own directory via `workspace: ${{ env.projectDir }}`.

Share a per-project directory with every sandbox launched from a global environment file, using the new projectDir variable.

```yaml
workspace: ${{ env.projectDir }}
```

— Exact variable names and config path given. v0.43.0

05 **Sandbox naming and conflict handling in sbx env** IMPROVED 62

Every `sbx env` subcommand gains a `--name` flag to override the sandbox name, and `sbx env run`, `sbx env create`, and `sbx env rm` now detect name conflicts with sandboxes created outside `sbx env`, providing guidance instead of treating them as environment-managed sandboxes.

— Flag and subcommands named, no example invocation. v0.43.0

---

06 **Deterministic verification of signed git kits** IMPROVED

60

Signed git kits now verify on every host by materializing checkouts from commit blobs alone, preventing smudge filters, LFS, hooks, and host git configuration from altering checked-out bytes.

— Clear mechanism but no command or flag to invoke. v0.43.0

THINNER COVERAGE BELOW

07 **Sandbox lifecycle visibility in ls and prune** IMPROVED

55

`sbx ls --json` output now includes a last-used timestamp, and `sbx prune` supports Docker-style `until` filtering.

— Named flags and output field, no example filter value. v0.43.0

---

08 **Immediate credential revocation via secret rm** IMPROVED

55

`sbx secret rm --sandbox` now immediately revokes the removed credential from the sandbox proxy.

— Named flag and effect, minimal further mechanism. v0.43.0

---

09 **Filesystem diagnostics for sbx diagnose** IMPROVED 54

`sbx diagnose` now checks whether `mkfs.erofs` is usable and warns if its default block size exceeds the sandbox kernel's page size.

— Specific check described, no command to run it shown. v0.43.0

---

10 **Registry mirror and cross-host auth support** NEW 52

Kits can now be installed through registry mirrors configured with an explicit port, and private registries can use an explicitly trusted cross-host authentication endpoint for sandbox pulls.

— Named surfaces but no config keys or commands shown. v0.43.0

11 **Windows client connectivity via AF_UNIX** NEW 41

Windows clients can now connect to `sandboxd` through filesystem `AF_UNIX` sockets.

— States the capability, no setup detail given. v0.43.0

12 **Faster sbx version command** IMPROVED 40

Plain `sbx version` now returns embedded version information without full CLI startup.

— Named command, no numbers on speed gain. v0.43.0

13 **Mount info in inspect commands** IMPROVED 35

`daemon inspect` and `inspect` now show mount information.

— Bare statement of new output, no field detail. v0.43.0

14 **Shallow Git repo support in clone-mode sandboxes** IMPROVED 32

Clone-mode sandboxes now support shallow Git repositories.

— One-line addition with no mechanism or command. v0.43.0

15 **Lower minimum sandbox memory** IMPROVED 30

The minimum memory for a sandbox has been decreased to 512 MiB.

— Single number, no config key or command shown. v0.43.0

└──▷ **BREAKING ON UPGRADE**

! `shareSkills` in `sbxenv.yaml` is replaced by `skills`, which accepts `off|readonly|readwrite`; existing `shareSkills` entries must be migrated to the new key.

! MCP OAuth client secrets are renamed from `mcp:<server>.client_secret` to `mcp:<server>:client_secret`; secrets stored under the old dotted name are no longer read and must be re-set with `sbx secret set mcp:<server>:client_secret`.

WAS THIS USEFUL? 👍 👎

PipeLab Pipelock

Feedback ⓘ 1 RELEASE · 2026-09-01 NOTES ↗

RANK

Pipelock is an open-source AI agent firewall that scans MCP and HTTP egress for exfiltration, SSRF, and prompt injection and emits signed action receipts.

> Pipelock v3.5.0 adds detection for current GitHub installation-token and Azure SAS token formats, native AEL evidence emission, and a set of operator-facing status and recorder improvements, alongside six breaking changes that tighten default security posture — including disabling npm lifecycle scripts by default, failing closed on unscannable HTTP trailers and MCP arguments, and blocking suppression of core DLP protections.

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

01 **Rules bundle schema, loading, and status warnings** NEW

65

Adds an exported `rules reader-schema` contract with atomic bundle loading and status publication, plus license-expiry warnings and rules-bundle compatibility warnings surfaced in operator status.

— Names contract and warning types; status surface gives starting point v3.5.0

02 **Emit config changes now require restart** BREAKING

65

Emit configuration changes now require a restart — reloads leave the active emitter unchanged and report the ignored change.

— Clear before/after behavior with restart as the remedy v3.5.0

03 **Core DLP and response floors can't be suppressed** BREAKING

65

Immutable core DLP and response floors can no longer be suppressed; existing suppressions naming those protected patterns must be removed or replaced with the documented scoped alternatives.

— Gives clear migration step though alternatives aren't named v3.5.0

04 **npm lifecycle scripts disabled by default** BREAKING

60

`npm` lifecycle scripts are disabled by default in contained environments; commands that require dependency install scripts must use the documented explicit override.

— Notes an override exists but doesn't name it v3.5.0

05 **Provider-key detection requires valid leading boundary**

60

BREAKING

Provider-key detection now requires a valid leading boundary; keys glued directly to a preceding key-alphabet character are no longer matched, removing some previously matched patterns.

— Precise mechanism change but no user-facing action v3.5.0

06 **Unscannable trailers and MCP args fail closed** BREAKING

60

Populated HTTP trailers that cannot be scanned and MCP tool arguments that redaction cannot safely inspect now fail closed rather than passing through.

— Clear before/after but no configuration to adjust it v3.5.0

THINNER COVERAGE BELOW

07 **GitHub installation-token and Azure SAS token detection** NEW

55

Adds detection and redaction for current `GitHub installation-token` and Azure SAS token formats.

— Names exact token formats but no config surface given v3.5.0

08 **External authority propagation at forwarding gates** NEW

55

Adds external authority propagation at forwarding gates, covering audit, emission, WebSocket, and MCP paths.

— Names all four covered paths but no config knob v3.5.0

09 **Offline recorder compaction and epoch inventory** NEW

55

Adds offline recorder compaction and legacy recorder-epoch inventory with bounded reads and integrity-preserving publication.

— Mechanism named but no command to invoke it v3.5.0

10 **Invalid duration values rejected in config validation**

40

BREAKING

Invalid duration values that overflow internal conversion are now rejected during configuration validation.

— Thin description, no affected config key named v3.5.0

11 **AEL evidence emission for session and activity records** NEW 35

Adds native AEL evidence emission for session lifecycle and activity records.

— AEL and its output format left unexplained v3.5.0

12 **Route-scoped entropy warnings** NEW 35

Adds route-scoped entropy warnings for selected request paths.

— No scoping mechanism or path syntax given v3.5.0

13 **Code-checked capability manifest** NEW 30

Adds a code-checked capability manifest that reports supported capabilities from attached implementation.

— Bare description with no access point named v3.5.0

└──▷ BREAKING ON UPGRADE

- ! npm lifecycle scripts are disabled by default in contained environments; commands that require dependency install scripts must use the documented explicit override.
- ! Emit configuration changes now require a restart — reloads leave the active emitter unchanged and report the ignored change.
- ! Provider-key detection now requires a valid leading boundary; keys glued directly to a preceding key-alphabet character are no longer matched, removing some previously matched patterns.
- ! Immutable core DLP and response floors can no longer be suppressed; existing suppressions naming those protected patterns must be removed or replaced with the documented scoped alternatives.
- ! Invalid duration values that overflow internal conversion are now rejected during configuration validation.
- ! Populated HTTP trailers that cannot be scanned and MCP tool arguments that redaction cannot safely inspect now fail closed rather than passing through.

WAS THIS USEFUL?  

// END OF TRANSMISSION

▷ Subscribe Feedback on this issue