THE AI TOOLCHAIN

NO. 040

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 28, 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 179 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 31 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. Braintrust's Patterns engine clusters recurring failures across traces so you stop reading logs one at a time. adds workload identity for external issuers, so an agent

vim ? help

authenticates as a subject under RBAC rather than sharing a key. And Grok API's `safety_identifier` pins a policy violation to the end user who caused it.

> OPERATE
>
> **Find the failure that keeps happening instead of reading traces one by one**
>
> Braintrust's Patterns engine surfaces recurring issues across traces, turning per-run inspection into a ranked list of what actually keeps breaking; its local secret-redaction plugins strip sensitive values before they leave your machine. Cotool lets agents act on alerts directly, so triage and remediation stop being two separate handoffs.
>
> Braintrust · Cotool

> GOVERN
>
> **Attribute an agent's actions to a subject, not to a shared API key**
>
> gram's workload identity API trusts external issuers and admits agent subjects under RBAC-gated scopes, which removes the practice of handing every agent the same long-lived credential. Grok API's `safety_identifier` carries an end-user handle on requests, so a policy violation points at the user who triggered it rather than at the key the whole app shares.
>
> gram · Grok API

> DEPLOY
>
> **Serve quantized and MoE models with visibility into why they are slow**
>
> LMDeploy exposes input/output logprobs on the serving API — the raw signal for confidence scoring and detecting degenerate outputs — and restructures MoE FFN execution for multi-node throughput. vMLX splits diagnostics into forward-call, wait, and CPU thread timings so you can tell a slow model from a starved scheduler; Darkbloom gates SSD prefix-cache writes on demand and extends cache lifetime, so a shared prompt prefix is recomputed less often.
>
> LMDeploy · vMLX

BUILD

◆ AI Coding Agents

Graphify Labs Graphify

Feedback ⓘ v0.9.71 NOTES ↗

RANK

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

> Graphify's latest releases expand knowledge-graph extraction to SQL triggers, Groovy enums, and JSX component relationships, while closing a Fortran path-traversal vulnerability and hardening Terraform secret redaction.

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

01 **Fortran preprocessing path-traversal fix** IMPROVED 90

Graphify strips every `#include` directive before the Fortran cpp preprocessing step and feeds source via stdin, closing a path-traversal that allowed `.F` / `.F90` files to inline arbitrary host files (including absolute paths like `/etc/passwd` and `../` traversals) into `graph.json`, `GRAPH_REPORT.md`, and the LLM context (GHSA-pcc4-rvhr-2pr8, CWE-22/73/200).

— Full CVE and mechanism detail; upgrade is the only action. v0.9.70

02 **SQL trigger extraction and table linking** NEW 75

Graphify extracts SQL `CREATE TRIGGER` statements and links them to their target table via `ON <table>`, supporting `OR REPLACE` / `OR ALTER`, `INSTEAD OF` triggers, and procedural `BEGIN…END` bodies.

— Names exact SQL syntax parsed but no runnable example. v0.9.71

03 **JSX component call edges for React graphs** NEW 75

Graphify adds `calls` edges for JSX component usage — `<MyButton/>` and `<_Row/>` link to their component definition, enabling React render-relationship traversal in the knowledge graph — while lowercase DOM tags (`<div>`) and member tags (`<Nav.Item>`) are conservatively skipped.

— Full mechanism and skip conditions but no runnable command. v0.9.70

04 **Terraform secret redaction for name/value pairs** IMPROVED

75

Graphify extends Terraform secret redaction to cover `value` fields paired with a secret-named `name` in name/value pair lists — e.g. `environment = [{ name = "DB_PASSWORD", value = "…" }]` — including ECS `valueFrom`, where the sensitive signal is the sibling `name` literal rather than a key.

— Concrete example given but no explicit config flag. v0.9.70

THINNER COVERAGE BELOW

05 **Groovy enum extraction** NEW

50

Graphify extracts Groovy `enum` declarations and their constants, linking members to the enum via `case_of` edges.

— Names edge type but lacks example or config detail. v0.9.71

v0.9.71 NOTES ↗ LATEST STABLE

Graphify v0.9.71 adds SQL trigger and Groovy enum extraction to the knowledge graph builder.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.9.71 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.71
```

Extracts SQL `CREATE TRIGGER` statements and links them to their target table via `ON <table>`, including `OR REPLACE` / `OR ALTER`, `INSTEAD OF`, and procedural `BEGIN…END` bodies.

Extracts Groovy `enum` declarations and their constants, with members linked to the enum via `case_of` edges.

v0.9.70 NOTES ↗ STABLE

Graphify v0.9.70 adds JSX `calls` edges for React component graphs and hardens Fortran, shell-injection, and Terraform secret-redaction paths.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.9.70 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.70
```

Adds `calls` edges for JSX component usage — `<MyButton/>` and `<_Row/>` link to the component definition, while lowercase DOM tags (`<div>`) and member tags (`<Nav.Item>`) are conservatively skipped, enabling React render-relationship traversal in the knowledge graph.

Strips every `#include` directive before the Fortran cpp preprocessing step and feeds source via stdin, closing a path-traversal that allowed `.F` / `.F90` files to inline arbitrary host files (including absolute paths like `/etc/passwd` and `../` traversals) into `graph.json` , `GRAPH_REPORT.md` , and the LLM context (GHSA-pcc4-rvhr-2pr8, CWE-22/73/200).

Extends Terraform secret redaction to cover `value` fields paired with a secret-named `name` in name/value pair lists — e.g. `environment = [{ name = "DB_PASSWORD", value = "…" }]` — including ECS `valueFrom` , where the sensitive signal is the sibling `name` literal rather than a key.

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.158.0 NOTES ↗ CODE ↗

RANK

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

> OpenAI Codex CLI's three releases add MCP OAuth client secret support, exec-server bearer-token auth and upstream proxy routing, transparent-background image generation, and a cluster of TUI, Guardian error-handling, and MCP status refinements.

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

01 exec-server bearer-token auth and upstream proxy routing NEW

90

Adds bearer-token authentication for direct exec-server WebSocket connections, including connections configured through app-server, and adds a `--proxy_private_ips_via_upstream` flag (env: `CODEX_EXEC_SERVER_PROXY_PRIVATE_IPS_VIA_UPSTREAM`) to the `exec-server` subcommand, allowing permitted private IP destinations to route through the configured upstream proxy instead of always connecting directly.

Set the proxy flag via environment variable in a deployment where CLI arguments are not convenient, such as a containerised exec-server.

```
$ CODEX_EXEC_SERVER_PROXY_PRIVATE_IPS_VIA_UPSTREAM=true codex exec-server
```

— Named flag, env var, and subcommand with runnable example rust-v0.158.0 rust-v0.159.0-alpha.12

02 Structured Guardian circuit-breaker errors NEW

85

Adds `auto_review.circuit_break_action = "strict"` config to attach structured `TooManyDenials` error info to interrupted turns when Guardian denial limits are hit, surfaced as `turn.error.codexErrorInfo = "tooManyDenials"` in notifications and saved history.

Opt in to structured Guardian circuit-breaker errors so denial-limit interruptions are identifiable in turn history and app-server notifications.

```yaml
auto_review:
  circuit_break_action: "strict"
```

— Named config key and exact error field with example `rust-v0.159.0-alpha.11`

03 **MCP OAuth client secret registration** NEW 75

Adds `codex mcp add --oauth-client-secret` to connect to MCP servers that require pre-registered OAuth client secrets, enabling automatic authentication with such servers.

Register an MCP server that requires a pre-registered OAuth client secret so Codex can authenticate automatically.

```
$ codex mcp add --oauth-client-secret <your-client-secret>
<server-url>
```

— Concrete command with runnable example, little mechanism detail `rust-v0.158.0`

04 **TUI copy, scrolling, and turn-duration display** IMPROVED 72

Adds `tui.copy_on_select` config key to control copy-on-select behavior in the fullscreen TUI, with `auto`, `always`, and `never` values (`auto` copies on mouse release unless the terminal is known to forward its native copy shortcut); enables mouse-wheel scrolling over the transcript while a modal dialog (such as 'Implement this plan?') is open; and shows short turn durations in TUI completion footers.

Set copy-on-select to always copy in terminals where the auto heuristic is too conservative.

```yaml
tui:
  copy_on_select: always
```

— Groups three thin TUI tweaks, names config key and values `rust-v0.158.0` `rust-v0.159.0-alpha.11`

05 **Single-server MCP status discovery** NEW 65

Adds an optional `serverName` parameter to `mcpServerStatus/list` for single-server MCP status discovery, reusing an existing thread connection and tool catalog when a `threadId` is provided.

— Named API parameter but no usage example `rust-v0.159.0-alpha.11`

THINNER COVERAGE BELOW

06 **Transparent backgrounds and file-backed images for image generation**

NEW 35

Image generation and editing can now explicitly request transparent backgrounds, and image edits now accept file-backed conversation images.

— Describes capability without flags, config, or examples `rust-v0.158.0`

07 **History-aware thread prewarming** IMPROVED 30

Adds history-aware prewarming for idle threads to reduce latency on subsequent turns.

— Brief description with no mechanism or config named `rust-v0.159.0-alpha.11`

08 **Elevated-permission commands require approval by default**

IMPROVED 25

Terminal input approval is now enabled by default for commands running with elevated permissions.

— Bare statement with no override mechanism given `rust-v0.158.0`

`rust-v0.158.0` NOTES ↗ CODE ↗ LATEST STABLE

Codex CLI v0.158.0 adds MCP OAuth client secrets, WebSocket bearer-token auth, transparent-background image generation, and configurable copy-on-select in the TUI.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.158.0
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.158.0
```

└──▷ TRY IT

Register an MCP server that requires a pre-registered OAuth client secret so Codex can authenticate automatically.

```
$ codex mcp add --oauth-client-secret <your-client-secret> <server-url>
```

Set copy-on-select to always copy in terminals where the auto heuristic is too conservative.

```yaml
tui:
  copy_on_select: always
```

Adds `codex mcp add --oauth-client-secret` to connect to MCP servers that require pre-registered OAuth client secrets.

Adds bearer-token authentication for direct exec-server WebSocket connections, including connections configured through app-server.

Adds `tui.copy_on_select` config key to control copy-on-select behavior in the fullscreen TUI, with `auto`, `always`, and `never` values; `auto` copies on mouse release unless the terminal is known to forward its native copy shortcut.

Enables terminal input approval by default for commands running with elevated permissions.

Image generation and editing can now explicitly request transparent backgrounds.

+1 more

`rust-v0.159.0-alpha.12` NOTES ↗ CODE ↗ PRE-RELEASE

Adds `proxy_private_ips_via_upstream` flag to `exec-server`, opt-in code-mode yielding on new user input, and item anchors for `thread/items/list` pagination.

└──▷ **GET THIS VERSION**

```
$ git clone --branch rust-v0.159.0-alpha.12
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.159.0-alpha.12
```

└──▷ **TRY IT**

Set the proxy flag via environment variable in a deployment where CLI arguments are not convenient, such as a containerised exec-server.

```
$ CODEX_EXEC_SERVER_PROXY_PRIVATE_IPS_VIA_UPSTREAM=true codex exec-server
```

Adds `--proxy_private_ips_via_upstream` flag (env: `CODEX_EXEC_SERVER_PROXY_PRIVATE_IPS_VIA_UPSTREAM`) to the `exec-server` subcommand, allowing permitted private IP destinations to route through the configured upstream proxy instead of always connecting directly.

`rust-v0.159.0-alpha.11` **NOTES ↗** **CODE ↗** **PRE-RELEASE**

Codex CLI adds single-server MCP status discovery, opt-in structured Guardian circuit-breaker errors, and history-aware thread prewarming.

└──▷ **GET THIS VERSION**

```
$ git clone --branch rust-v0.159.0-alpha.11 https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.159.0-alpha.11
```

└──▷ **USE IT**

Opt in to structured Guardian circuit-breaker errors so denial-limit interruptions are identifiable in turn history and app-server notifications.

```yaml
auto_review:
  circuit_break_action: "strict"
```

Adds `auto_review.circuit_break_action = "strict"` config to attach structured `TooManyDenials` error info to interrupted turns when Guardian denial limits are hit, surfaced as `turn.error.codexErrorInfo = "tooManyDenials"` in notifications and saved history.

Adds optional `serverName` parameter to `mcpServerStatus/list` for single-server MCP status discovery, reusing an existing thread connection and tool catalog when a `threadId` is provided.

Adds history-aware prewarming for idle threads to reduce latency on subsequent turns.

Enables mouse-wheel scrolling over the transcript while a modal dialog (such as 'Implement this plan?') is open.

Shows short turn durations in TUI completion footers.

WAS THIS USEFUL?  

Letta Code

Feedback ⓘ v0.33.5 NOTES ↗ RANK

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

> Letta Code added a GitHub PR monitoring tool, fresh-agent MemFS context, automatic backgrounding of slow tools, and workflow engine tuning across two releases.

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

01 **WatchPR tool for GitHub PR monitoring** NEW 50

Adds a `WatchPR` tool to monitor GitHub pull requests from within agents.

— Names the tool but gives no usage detail v0.33.3

02 **Auto-backgrounding of slow external tools** IMPROVED 50

Auto-backgrounds external tools that run longer than ten seconds, keeping the agent responsive [LET-13311].

— Specific threshold and ticket ref, no toggle named v0.33.3

03 **Root MemFS for fresh local agents** NEW 45

Creates fresh local agents with a root MemFS, giving new agents a clean git-tracked context from first launch.

— Explains behavior but no config or command shown v0.33.3

04 **Cloud sandbox disk space warnings** NEW 30

Warns agents proactively when their Cloud sandbox is running low on disk space.

— Brief description, no threshold or config given v0.33.3

05 **Workflow engine decision calibration and worker memory guard** IMPROVED 30

Adds calibrated Jev decisions to the workflow engine for more precise agent decision-making, and disables the memory guard for workflow worker agents, allowing workers to operate without memory constraints.

— Vague on mechanism, no concrete configuration named v0.33.5

v0.33.5 NOTES ↗ LATEST STABLE

Letta Code v0.33.5 adds calibrated Jev decisions and disables memory guard for workflow workers.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.33.5 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.33.5
```

Adds calibrated Jev decisions to the workflow engine for more precise agent decision-making.

Disables memory guard for workflow worker agents, allowing workers to operate without memory constraints.

v0.33.3 NOTES ↗ STABLE

Adds WatchPR for GitHub PR monitoring, root MemFS for fresh local agents, and auto-backgrounding of slow external tools.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.33.3 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.33.3
```

Adds `WatchPR` tool to monitor GitHub pull requests from within agents.

Creates fresh local agents with a root MemFS, giving new agents a clean git-tracked context from first launch.

Auto-backgrounds external tools that run longer than ten seconds, keeping the agent responsive [LET-13311].

Warns agents proactively when their Cloud sandbox is running low on disk space.

WAS THIS USEFUL? 👍 👎

Cotool

Feedback ⓘ v0.71.0 NOTES ↗ RANK

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

> Cotool v0.71.0 lets agents act directly on alerts, adds sandbox network controls, delivery tracking with resend, and expands investigation data sources, while removing the per-agent severity filter so all triggered results are delivered.

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

01 **Cotool Alerts in agent tool picker** NEW 75

Adds Cotool Alerts to the agent tool picker so any agent can search, read, comment on, update, and create alerts as part of its investigation steps. Enable it via the agent tool picker to let an agent triage and update alerts end-to-end.

Enable an agent to triage and update alerts end-to-end without leaving the agent workflow — useful for auto-escalation or enrichment pipelines.

📍 In the agent tool picker, enable Cotool Alerts. The agent can now call search, read, comment on, update, and create alerts as part of its investigation steps.

— Includes exact enable steps and named actions v0.71.0

02 **Agent run and notification cards on alert page** NEW 65

Adds agent run and notification cards to the alert detail page, linking directly to the Slack thread, Linear issue, Tines case, or PagerDuty incident each escalation created.

— Names four integrations but no navigation steps v0.71.0

03 **Per-agent minimum severity filter removed** BREAKING 60

Agents now deliver every triggered result; the per-agent minimum severity filter that could previously hide lower-severity results has been removed, which may increase alert volume for destinations that relied on that filter.

— Explains behavior change and downstream impact v0.71.0

THINNER COVERAGE BELOW

04 **Sandbox network controls** NEW 55

Adds Sandbox network settings so administrators control which hosts and ports agent sandboxes can reach.

— Names the setting but not exact values or location v0.71.0

05 **Delivery tracking for agent runs** NEW 55

Adds delivery tracking to agent runs, showing what each run sent and whether it succeeded, with one-click resend for failed deliveries.

— Describes the feature but not exact UI path v0.71.0

06 **Agent investigation data sources expanded** IMPROVED 55

Improves agent investigations with dismissed and expired alerts now included in alert searches, Slack user lookup by email, Wiz Blue Agent analysis retrieval, and Code42 watchlist membership lookups.

— Lists four data sources without config or command detail v0.71.0

07 **MITRE ATT&CK coverage updated to v19.2** IMPROVED 45

Updates MITRE ATT&CK coverage to v19.2, with new ATT&CK releases now applying automatically going forward.

— Version bump noted; no manual action required v0.71.0

08 **New LLM models added to catalog** NEW 40

Adds Claude Opus 5.5, GPT-6 Sol, and GPT-6 Luna to the model catalog.

— Names models with no usage guidance v0.71.0

09 **AWS CLI assume-role authentication for EU** NEW 35

Adds AWS CLI assume-role authentication for EU deployments.

— Bare mention with no setup detail v0.71.0

└──▷ **BREAKING ON UPGRADE**

! The per-agent minimum severity filter has been removed; agents now deliver every triggered result regardless of severity, which may increase alert volume for destinations that previously relied on that filter.

v0.71.0 NOTES ↗ LATEST STABLE

Cotool v0.71.0 adds sandbox network controls, Cotool Alerts in the agent tool picker, new LLM models, and AWS CLI assume-role auth for EU deployments.

└──▷ **HOW TO FIND IT**

Enable an agent to triage and update alerts end-to-end without leaving the agent workflow — useful for auto-escalation or enrichment pipelines.

📍 In the agent tool picker, enable `Cotool Alerts`. The agent can now call search, read, comment on, update, and create alerts as part of its investigation steps.

Adds `Sandbox network settings` so administrators control which hosts and ports agent sandboxes can reach.

Adds Cotool Alerts to the agent tool picker so any agent can search, read, comment on, update, and create alerts.

Adds AWS CLI assume-role authentication for EU deployments.

Adds Claude Opus 5.5, GPT-6 Sol, and GPT-6 Luna to the model catalog.

Adds delivery tracking to agent runs, showing what each run sent and whether it succeeded, with one-click resend for failed deliveries.

+4 more

└──▷ **BREAKING ON UPGRADE**

! The per-agent minimum severity filter has been removed; agents now deliver every triggered result regardless of severity, which may increase alert volume for destinations that previously relied on that filter.

WAS THIS USEFUL? 👍 👎

Worktrunk

Feedback ⓘ v0.80.0 NOTES ↗ RANK

Worktrunk manages Git worktrees from the command line for parallel AI coding-agent workflows, with hooks, status listings, and LLM-written commits.

> Worktrunk sped up CI status checks in large repos, added Sigstore-signed release manifests, refined Codex-driven commit generation, and introduced a scoped worktree-prune subcommand — alongside a breaking change to a JSON log field.

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

01 Real branch names in worktrunk log JSON output

BREAKING

85

The `branch` field returned by `wt config state logs --format=json` now contains the real branch name (e.g. `feature/x`) or `null`, whereas previously it held the sanitised log-directory name (e.g. `feature-x-x2d`); filters written against the sanitised form must be updated.

— Exact command, before/after field values, migration note. v0.80.0

02 Codex commit instruction file and model tuning

IMPROVED

83

Accepting the first-run Codex prompt creates `~/.codex/worktrunk-commit-instructions.txt`, which the Codex command reads to customise commit generation; it must be removed with `rm ~/.codex/worktrunk-commit-instructions.txt` after changing or deleting the saved commit command. Codex commit generation also now recommends the `gpt-6-luna` model with reasoning disabled and unneeded features turned off, sending fewer input tokens per message.

Reset Codex commit instructions after updating or removing the saved commit command so the stale file does not cause failures.

```
$ rm ~/.codex/worktrunk-commit-instructions.txt
```

— Names file path, exact rm command, and model change. v0.80.0

03 Scoped stale-worktree prune subcommand

NEW

80

Adds `wt step prune` to unregister a stale worktree by deleting its `.git/worktrees/<id>` entry, scoped to that single worktree only (unlike `git worktree prune`), while preserving files in the working directory.

Unregister a single stale worktree that no longer exists on disk without pruning all worktrees globally.

```
$ wt step prune
```

— Names exact command, mechanism, and contrast with git's own prune. product docs

04 **Faster CI status checks in wt list and statusline** IMPROVED

80

CI status queries in `wt list --full`, the picker, and the statusline now skip branches that have never been pushed to a remote, reducing `gh` calls from 249 to 25 on a repo with 239 local branches; GitLab project ID is also looked up once per command instead of once per row.

— Concrete before/after numbers and named surfaces. v0.80.0

05 **Sigstore-signed release manifests** NEW

65

Each GitHub release now includes `packslip.sigstore.json`, a Sigstore-signed manifest of archive digests, platforms, completions, and agent skills, enabling installers such as mise to verify downloads before unpacking.

— Names manifest file and use, but no verification command shown. v0.80.0

└──▷ **BREAKING ON UPGRADE**

! The `branch` field returned by `wt config state logs --format=json` now contains the real branch name (e.g. `feature/x`) or `null`; previously it held the sanitised log-directory name (e.g. `feature-x-x2d`). Any filters written against the sanitised form must be updated.

v0.80.0 NOTES ↗ LATEST STABLE

Worktrunk v0.80.0 speeds up `wt list` / `wt switch` in large repos and ships signed release manifests with `packslip.sigstore.json`.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.80.0 https://github.com/max-sixty/worktrunk.git
# already have the repo? check out this version:
$ git checkout v0.80.0
```

Accepting the first-run Codex prompt now creates `~/.codex/worktrunk-commit-instructions.txt`, which the Codex command reads to customise commit generation.

Each GitHub release now includes `packslip.sigstore.json`, a Sigstore-signed manifest of archive digests, platforms, completions, and agent skills, enabling installers such as mise to verify downloads before unpacking.

CI status queries in `wt list --full`, the picker, and the statusline now skip branches that have never been pushed to a remote, reducing `gh` calls from 249 to 25 on a repo with 239 local branches; GitLab project ID is also looked up once per command instead of once per row.

Codex commit generation updates the recommended model to `gpt-6-luna` with reasoning disabled and unneeded features turned off, sending fewer input tokens per message.

└─▷ **BREAKING ON UPGRADE**

! The `branch` field returned by `wt config state logs --format=json` now contains the real branch name (e.g. `feature/x`) or `null`; previously it held the sanitised log-directory name (e.g. `feature-x-x2d`). Any filters written against the sanitised form must be updated.

WAS THIS USEFUL? 👍 👎

mex

Feedback ⓘ v0.8.3 CODE ↗ RANK

Mex indexes codebases through a code graph and supplies a repo-local Markdown wiki to AI coding agents.

> Mex 0.8.3 adds route resolvers for FastAPI, Flask, and NestJS, pyproject.toml dependency checks, a scaffold export command, and Markdown timelines, alongside new Hub Graph/Wiki commands and setup flags — but upgrading changes TypeScript node IDs and requires a `mex graph refresh` before `mex sync`.

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

01 Standalone skill installation via skills.sh NEW 85

`npx skills add mex-memory/mex` installs MEX agent skills as an alternative to the built-in MEX skill installer. Running it over a project already managed by `mex setup` or `mex skills sync` replaces MEX's managed directories, causing later setup or sync operations to report ownership or path conflicts — pick one installer per agent integration.

Install MEX agent skills into an existing project without re-running the full MEX installer.

```
$ npx skills add mex-memory/mex
```

— Exact command plus concrete conflict behavior to avoid. product docs

02 FastAPI route resolver NEW 85

A new FastAPI route resolver connects @app.get(), @router.post(), @app.api_route() (with `methods=` list), and other HTTP shortcut decorators to their handler functions, emitting one route node per declared method with edges labeled `fastapi-route-handler`.

— Names exact decorators and graph edge label used. v0.8.3

03 Flask route resolver NEW 85

A new Flask route resolver recognizes @app.route(), shortcut decorators (@app.get(), @app.post(), @app.put(), @app.patch(), @app.delete(), @app.options(), @app.head()), and Blueprint instances, fanning multi-method declarations into one stable route node per method.

— Names exact decorators and Blueprint handling. v0.8.3

04 NestJS controller route resolver NEW 85

A new NestJS controller route resolver detects `@nestjs/core` or `@nestjs/common` dependencies and recognizes @Controller() prefixes with @Get(), @Post(), @Put(), @Patch(), @Delete(), @Options(), @Head(), and @All() method decorators.

— Names exact decorators and dependency detection logic. v0.8.3

---

05 **TypeScript node ID migration on upgrade** BREAKING 85

Upgrading to 0.8.3 changes TypeScript node IDs after the first full extraction/graph refresh; run `mex graph refresh` before `mex sync` (e.g. `mex graph refresh && mex sync && mex wiki rebuild-index`) so MEX can map old IDs to their replacements, then manually review any ambiguous or missing groundings.

Before syncing after a 0.8.3 upgrade, refresh the graph incrementally so MEX can reconcile changed TypeScript node IDs.

```
$ mex graph refresh && mex sync && mex wiki rebuild-index
```

— Gives exact migration command sequence to run. product docs

---

06 **Setup command flags for headless, CLI-only, and agent-memory modes**

NEW 80

`npx mex-agent@0.8.3 setup` gains `--no-open` to print the local browser link without launching a browser, `--cli` for CLI-only setup mode, and `--mode agent-memory` for agent-memory installation mode.

Set up MEX in headless/CI environments where opening a browser is not possible.

```
$ npx mex-agent@0.8.3 setup --no-open
```

Set up MEX in a CI or headless environment where opening a browser is not possible.

```
$ npx mex-agent@0.8.3 setup --no-open
```

— Names three exact flags with runnable examples. product docs

07 **Dependency checks against pyproject.toml** NEW 80

Dependency claims are now checked against `pyproject.toml`, scanning `[project] dependencies`, `[project.optional-dependencies]` per-extra arrays, PEP 735 `[dependency-groups]`, and `[tool.poetry.dependencies]` with named groups.

— Names exact config sections scanned. v0.8.3

08 **New Hub Graph and Wiki index commands** NEW 65

Adds `npx mex-agent@0.8.3 hub` subcommand, `npx mex-agent@0.8.3 graph rebuild` for full graph reconstruction, and `npx mex-agent@0.8.3 wiki rebuild-index` to rebuild the Wiki index.

— Three commands named but no further behavior described. product docs

09 **Markdown-formatted timeline export** NEW 65

`mex timeline` gains a `--format md` flag producing Markdown-formatted timeline output, e.g. `mex timeline --format md`, useful for sharing project history or committing to a repo.

Export a Markdown timeline to share project memory history with teammates or include in docs.

```
$ mex timeline --format md
```

Export a project timeline as a Markdown file for sharing or committing to a repo.

```
$ mex timeline --format md
```

Generate a Markdown-formatted timeline for inclusion in reports or documentation.

```
$ mex timeline --format md
```

— Exact flag with runnable example provided. v0.8.3

THINNER COVERAGE BELOW

10 **Visibility into stale and unindexed files** IMPROVED 55

The `scope` command now surfaces text-only bounded live-text evidence for stale or unindexed files, clearly labelled as such, and graph output now surfaces unindexed source files instead of silently omitting them.

— Names `scope` command and graph output change, no exact syntax. v0.8.3

11 **Incremental Hub Graph refresh** IMPROVED 50

Hub Graph refresh now reuses unchanged extractions and rewrites only changed rows, instead of performing a full extraction every time.

— Describes before/after mechanism but no command surface. product docs

12 **Scaffold export command** NEW 50

`mex export` bundles the scaffold into a single Markdown file.

— Names exact command but limited description of contents. v0.8.3

13 **Heartbeat staleness config changes** IMPROVED 45

Heartbeat now surfaces inactive staleness checks on legacy scaffolds, and day-based heartbeat thresholds in config accept zero as a valid value.

— Describes config behavior but no exact config key given. v0.8.3

14 **Disposable process for Hub Graph construction** IMPROVED 35

Hub Graph construction now runs in a disposable process so compiler memory is released when the job exits.

— States mechanism but no user-facing action. product docs

└──▷ **BREAKING ON UPGRADE**

! Upgrading to 0.8.3 changes TypeScript node IDs after the first full extraction; run `mex graph refresh` before `mex sync` to allow MEX to map old IDs to their replacements, then manually review ambiguous or missing groundings.

! Upgrading to 0.8.3 changes TypeScript node IDs on first graph refresh; run `mex graph refresh` before `mex sync` to allow MEX to map old IDs to replacements, then re-review any ambiguous or missing groundings.

! Using `npx skills add mex-memory/mex` (skills.sh) over a project already managed by `mex setup` or `mex skills sync` replaces MEX's managed directories and causes later setup or sync operations to report ownership or path conflicts — choose one installer per agent integration.

v0.8.3 CODE ↗ LATEST STABLE

mex v0.8.3 adds scaffold export, Markdown timelines, Flask/FastAPI/NestJS route resolvers, and pyproject.toml dependency checks

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.8.3 https://github.com/mex-memory/mex.git
# already have the repo? check out this version:
$ git checkout v0.8.3
```

└──▷ **TRY IT**

Generate a Markdown-formatted timeline for inclusion in reports or documentation.

```
$ mex timeline --format md
```

Adds `mex export` command to bundle the scaffold into a single Markdown file.

Adds `--format md` flag to `mex timeline` for Markdown-formatted timeline output.

Adds a FastAPI route resolver connecting @app.get(), @router.post(), @app.api_route() (with `methods=` list), and other HTTP shortcut decorators to their handler functions, emitting one route node per declared method with edges labeled `fastapi-route-handler`.

Adds a Flask route resolver recognizing @app.route(), shortcut decorators (@app.get(), @app.post(), @app.put(), @app.patch(), @app.delete(), @app.options(), @app.head()), and Blueprint instances, fanning multi-method declarations into one stable route node per method.

Adds a NestJS controller route resolver detecting `@nestjs/core` or `@nestjs/common` dependencies and recognizing @Controller() prefixes with @Get(), @Post(), @Put(), @Patch(), @Delete(), @Options(), @Head(), and @All() method decorators.

+4 more

WAS THIS USEFUL? 👍 👎

StackBlitz bolt.new

Feedback ⓘ snapshot-20260928 NOTES ↗

RANK

Bolt.new is an AI-powered web development platform that builds websites and applications from prompts.

> Bolt.new added a Model training privacy setting letting users control whether their projects and chats are used to train AI models.

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

01 **Model training opt-out setting** NEW 80

A new Model training toggle in Account Settings lets users control whether Bolt may use their projects and chats to train AI models. It defaults to off for EU, UK, and Switzerland accounts and on for all others, effective for content created on or after October 7, 2026; users can navigate to Account Settings > Model training to toggle it off.

Opt out of AI model training for your account so projects and chats submitted on or after October 7, 2026 are not used for training.

> 📍 In Bolt, go to Account Settings › Model training and toggle the setting off.

Opt out of AI model training on your content — important for teams handling sensitive code or data.

> 📍 In Bolt, go to Account Settings › Model training and toggle the setting off.

— Names exact setting, defaults, regions, and effective date snapshot-20260928

snapshot-20260928 NOTES ↗ LATEST STABLE

Bolt adds a Model training setting so you can control whether your projects and chats are used to train AI models.

└──▷ **HOW TO FIND IT**

Opt out of AI model training for your account so projects and chats submitted on or after October 7, 2026 are not used for training.

> 📍 In Bolt, go to Account Settings › Model training and toggle the setting off.

WAS THIS USEFUL? 👍 👎

Vercel v0

Feedback ⓘ snapshot-20260928 NOTES ↗ RANK

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

> v0 shipped new MCP server tools for reading chat history and resolving tasks, an Enterprise AI Gateway model picker with custom model access, configurable default chat visibility for teams, team-scoped Design Systems, and activity indicators on project cards.

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

01 Enterprise AI Gateway model picker NEW

80

Adds an Allow custom models toggle in Workspace settings for Enterprise team owners, enabling AI Gateway model access including token pricing review and per-model Thinking and Fast Mode customization via the updated model picker in the chat composer.

Allow Enterprise team members to browse and use AI Gateway models beyond the default v0 models, including controlling Thinking and Fast Mode per model.

📍 In v0, go to Workspace settings and enable Allow custom models, then have team members open the chat composer, click Add models, and select from available AI Gateway models.

— Named toggle and navigation steps given, mechanism explained snapshot-20260928

02 Default chat visibility setting NEW

78

Adds Default Chat Visibility setting under Settings → Workspace → Default Chat Visibility with three options — Private, Team can view, and Team can edit — controlling visibility of new team workspace chats.

Set new team workspace chats to be visible and editable by all team members by default, so collaborators can contribute without waiting for an invite.

📍 In v0, go to Settings → Workspace → Default Chat Visibility and select Team can edit.

— Exact setting path and option values with usage steps snapshot-20260928

03 MCP server chat history and task tools NEW

75

Adds `listMessages` and `resolveTask` tools to the v0 MCP server, enabling programmatic reading of chat history and answering plan approvals, questions, permission requests, and integration prompts; the server now returns scope-aware chat URLs.

— Names exact MCP tools but no example call shown snapshot-20260928

THINNER COVERAGE BELOW

04 **Team-scoped Design Systems default** NEW 45

Adds team-scoped Design Systems default, including Design Systems 2.0 skills and built-in examples, with scope selection (team or personal) when saving from the save card.

— Names feature and scope option but no concrete steps snapshot-20260928

05 **Teammate activity on project cards** IMPROVED 30

Project cards now surface teammates active in visible project chats within the last 24 hours, ordered by most recent activity.

— Thin UI description with no named surface or steps snapshot-20260928

snapshot-20260928 NOTES ↗ LATEST STABLE

v0 adds Enterprise AI Gateway model picker, configurable team chat visibility defaults, and MCP server tools for chat history and task resolution.

└──▷ HOW TO FIND IT

Set new team workspace chats to be visible and editable by all team members by default, so collaborators can contribute without waiting for an invite.

📍 In v0, go to Settings → Workspace → Default Chat Visibility and select Team can edit.

Allow Enterprise team members to browse and use AI Gateway models beyond the default v0 models, including controlling Thinking and Fast Mode per model.

📍 In v0, go to Workspace settings and enable Allow custom models, then have team members open the chat composer, click Add models, and select from available AI Gateway models.

Adds `listMessages` and `resolveTask` tools to the v0 MCP server, enabling programmatic reading of chat history and answering plan approvals, questions, permission requests, and integration prompts; the server now returns scope-aware chat URLs.

Adds Allow custom models toggle in Workspace settings for Enterprise team owners, enabling AI Gateway model access including token pricing review and per-model Thinking and Fast Mode customization via the updated model picker in the chat composer.

> Adds `Default Chat Visibility` setting under `Settings → Workspace → Default Chat Visibility` with three options — `Private`, `Team can view`, and `Team can edit` — controlling visibility of new team workspace chats.
>
> Adds team-scoped Design Systems default, including Design Systems 2.0 skills and built-in examples, with scope selection (team or personal) when saving from the save card.
>
> Project cards now surface teammates active in visible project chats within the last 24 hours, ordered by most recent activity.

WAS THIS USEFUL?  

Anysphere Cursor

Feedback ⓘ snapshot-20260928 NOTES ↗

RANK

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

> Cursor introduces two new PR bots — Rollouts for deploy-health monitoring and Security Review for exploitable-bug detection — both rolled out to Teams and Enterprise plans with introductory usage credits.

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

01 **Rollouts bot for PR deploy-health monitoring** NEW 88

Enabled from the dashboard's Automations tab, Rollouts attaches a monitor to every pull request and reports change health per environment as verified healthy, regression detected, or inconclusive, by running a generated monitoring plan against logs, metrics, and traces. When a PR opens, Rollouts posts the monitoring plan as a PR comment listing identified risks, intended effect, signals to check, and instrumentation gaps; editing the comment causes Rollouts to use the revised plan. It integrates with Origin or GitHub for source control, continuous delivery systems for deploy events, and Datadog and other telemetry providers for signals.

Enable Rollouts so every future PR in your repo gets a deploy-health monitor connected to GitHub and Datadog.

> 📍 In the Cursor dashboard, go to Automations tab › Enable Rollouts › Connect source control (Origin or GitHub), your continuous delivery system, and your telemetry provider (e.g. Datadog). Rollouts begins monitoring on the next pull request opened.

— Names states, plan contents, and integrations; UI path only, no exact flag.

snapshot-20260928

THINNER COVERAGE BELOW

02 **Introductory credits for Rollouts and Security Review** NEW 30

Rollouts and Security Review are available on Teams and Enterprise plans; Teams customers receive credits for roughly 50 changes and Enterprise for roughly 500 changes during the introductory period.

— States plan tiers and rough credit counts, no mechanism or action. snapshot-20260928

└──▷ **ALSO FROM THESE RELEASES**

snapshot-20260928 NOTES ↗ LATEST STABLE

Cursor launches Rollouts (deploy health monitoring) and Security Review (exploitable-bug detection) bots for every PR.

└──▷ **HOW TO FIND IT**

Enable Rollouts so every future PR in your repo gets a deploy-health monitor connected to GitHub and Datadog.

> 📍 In the Cursor dashboard, go to Automations tab › Enable Rollouts › Connect source control (Origin or GitHub), your continuous delivery system, and your telemetry provider (e.g. Datadog). Rollouts begins monitoring on the next pull request opened.

Enable Security Review to automatically flag exploitable bugs (injection, auth bypasses, committed secrets, SSRF, unsafe deserialization) on every non-draft PR in selected repositories.

> 📍 In the Cursor dashboard, go to Automations tab › Enable Security Review › Select the repositories you want reviewed. Draft PRs are skipped automatically; all other PRs receive one review comment per scan.

Adds Rollouts bot, enabled from the dashboard's automations tab, which attaches a monitor to every pull request and reports change health per environment (verified healthy, regression detected, or inconclusive) by running a generated monitoring plan against logs, metrics, and traces.

Rollouts generates a monitoring plan as a PR comment when a pull request opens, listing identified risks, intended effect, signals to check, and instrumentation gaps; editing the comment causes Rollouts to use the revised plan.

Rollouts integrates with Origin or GitHub for source control, continuous delivery systems for deploy events, and Datadog and other telemetry providers for signals.

Security Review supports team-defined codebase rules (e.g. which client external calls must go through a specific path, or which tables are never queried from a request handler) enforced on every PR.

Rollouts and Security Review are available on Teams and Enterprise plans; Teams customers receive credits for roughly 50 changes and Enterprise for roughly 500 changes during the introductory period.

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

Models: Claude Opus 5.5 Now Available: Claude Opus 5.5 brings stronger agentic coding an

BLOG ↗ RANK

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

> Kiro added support for Claude Opus 5.5 across its IDE, CLI, Crew, and Web surfaces, offering stronger agentic coding at a lower credit cost than its predecessor.

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

01 **Claude Opus 5.5 model support** NEW 75

Claude Opus 5.5 is now available in Kiro IDE, CLI, Crew, and Web, offering stronger agentic coding and clearer communication than Opus 5 at a 2.0x credit multiplier (down from Opus 5's 2.2x) with 1M-token context. To pick it up, restart the IDE or CLI, run `kirocrew restart` for Crew, or refresh Web in the browser.



Switch Kiro Crew to use Claude Opus 5.5 after the model list is updated.

```
$ kirocrew restart
```

— Names model, multiplier, context size, and exact refresh commands.

Models: Claude Opus 5.5 Now Available: …

Models: Claude Opus 5.5 Now Available: Claude Opus 5.5 brings stronger agentic coding

BLOG ↗ LATEST STABLE

Claude Opus 5.5 now available in Kiro IDE, CLI, Crew, and Web with 1M-token context at a 2.0x credit multiplier.

└──▷ **TRY IT**

Switch Kiro Crew to use Claude Opus 5.5 after the model list is updated.

```
$ kirocrew restart
```

Refreshes the available model list by restarting the IDE or CLI, running `kirocrew restart` for Crew, or refreshing Web in the browser.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

EXXETA exxperts

Feedback ⓘ v0.14.0 NOTES ↗ RANK

The exxperts framework runs local-first AI agents with approval-gated memory, model-provider integrations, MCP tools, and web search.

> exxperts v0.14.0 introduces per-room AI model selection with mid-conversation switching and defaults, adds synced room sorting, and rounds out chat UX with a streaming scroll pill, longer Remember processing, auto-reconnection, keyboard shortcuts, and a redesigned settings layout.

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

01 **Per-room model selection and defaults** NEW 60

Room settings now let you pick separate models for conversations and for memory work in each room, with the Home card showing the active model and selections persisting across Remember and Forget. You can also switch the model mid-conversation and it continues on the newly chosen model, and AI setup lets you configure per-room model defaults that act as the fallback for new rooms and any room without its own selection (existing rooms keep their current models).

Set a dedicated model for a room's memory work so long-context summarization uses a different model than day-to-day chat.

📍 In a room, go to Room settings › Model and choose one model for conversations and a separate model for memory work, then save.

— Names Room settings, Home card, Remember/Forget, AI setup but no config keys v0.14.0

THINNER COVERAGE BELOW

02 **Keyboard shortcuts for chat confirmations** NEW 55

Enter and Escape now answer yes/no questions and confirmations in chat, and destructive dialogs default focus to Cancel so pressing Enter alone never removes data.

— Names exact keys and safety behavior, no example shown v0.14.0

03 **Room sorting with cross-device sync** NEW 50

Rooms can now be sorted by last used, by name, or by custom drag order, and the chosen order syncs across every device.

Re-order your rooms to surface the most-used ones without relying on recency or alphabetical sort.

📍 On the Home screen, open the sort controls and drag rooms into your preferred order — the order syncs to all your devices automatically.

— Names three sort modes and gives UI steps, lacks deeper mechanism v0.14.0

04 **Long conversation handling in Remember** IMPROVED 45

Remember now supports conversations of any length by reading long conversations in parts, with a cancel option available during processing.

— States mechanism (reading in parts) but no limits or numbers v0.14.0

05 **'New text' pill during streaming** NEW 40

A New text pill appears when you scroll up while an answer is streaming, letting you jump back to the latest text without losing your current scroll position.

— Describes behavior but no navigation path or example given v0.14.0

06 **Automatic reconnection with Reconnect button** IMPROVED 40

When the connection drops, the app now automatically reconnects and shows a Reconnect button, while preserving any unsaved typed text.

— Names the Reconnect button but no further detail on triggers v0.14.0

07 **Redesigned Settings, Room settings, and Skills UI** IMPROVED 40

Settings, Room settings, and Skills were redesigned so every setting occupies one row, and closing a panel with unsaved changes now prompts to Save and close.

— Names three redesigned screens but describes layout only generically v0.14.0

v0.14.0 NOTES ↗ LATEST STABLE

exxperts v0.14.0 adds per-room model selection, room sorting, and a New text scroll pill for streaming chats.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.14.0 https://github.com/EXXETA/exxperts.git
# already have the repo? check out this version:
$ git checkout v0.14.0
```

└──▷ HOW TO FIND IT

Set a dedicated model for a room's memory work so long-context summarization uses a different model than day-to-day chat.

📍 In a room, go to Room settings › Model and choose one model for conversations and a separate model for memory work, then save.

Re-order your rooms to surface the most-used ones without relying on recency or alphabetical sort.

📍 On the Home screen, open the sort controls and drag rooms into your preferred order — the order syncs to all your devices automatically.

Adds per-room model selection in Room settings, with separate model choices for conversations and memory work; the Home card displays the active model and selections persist across Remember and Forget.

Enables switching the AI model mid-conversation between answers, so the conversation continues on the newly chosen model.

Adds room sorting by last used, by name, or by custom drag order — synced across every device.

Adds a `New text` pill that appears when scrolling up during a streaming answer, letting you jump back to the latest text without losing your scroll position.

Adds per-room model defaults via AI setup, which sets the fallback model for new rooms and any room without its own selection; existing rooms keep their current models.

+4 more

WAS THIS USEFUL? 👍 👎

Feynman

Feedback ⓘ v0.5.11 NOTES ↗ RANK

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

> Feynman added a Crossref polite-pool opt-in with clearer rate-limit errors during citation verification.

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

01 **Crossref polite pool via CROSSREF_MAILTO** IMPROVED 80

Setting the `CROSSREF_MAILTO` environment variable to your email opts Feynman into Crossref's faster polite pool for citation verification, and surfaces a specific, actionable message on 429 rate-limit responses instead of a generic error.

Avoid 429 errors during parallel DOI citation checks by opting into Crossref's polite pool via your email address.

```
$ export CROSSREF_MAILTO=you@example.com
feynman deepresearch "scaling laws in large language models"
```

— Names exact env var and shows runnable command. v0.5.11

v0.5.11 NOTES ↗ LATEST STABLE

Set `CROSSREF_MAILTO` to unlock Crossref's polite pool and get clearer 429 error messages during citation verification.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.5.11 https://github.com/Companion-
Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.11
```

└──▷ **TRY IT**

Avoid 429 errors during parallel DOI citation checks by opting into Crossref's polite pool via your email address.

```
$ export CROSSREF_MAILTO=you@example.com
feynman deepresearch "scaling laws in large language models"
```

> Setting `CROSSREF_MAILTO` to your email opts Feynman into Crossref's faster polite pool and surfaces a specific, actionable message on 429 rate-limit responses instead of a generic error.

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.12.0 NOTES ↗ RANK

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

> Gram's biggest addition this window is a workload identity management API and matching admin page for trusting external issuers and admitting agent subjects under RBAC-gated scopes, alongside an AI-ranked command palette (launcher.judge), Slack workspace connection with directory sync, MCP-scoped risk policies, and directory role mappings for SSO groups.

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

01 AI-ranked command palette via launcher.judge NEW 90

Adds the `launcher.judge` management endpoint (`POST /rpc/launcher.judge`) that routes command-palette intent decisions to Jev via OpenRouter, returning probability distributions keyed by candidate IDs; organizations without an OpenRouter key receive `disabled: true` . The dashboard command palette (⌘K) sends typed text and prefiltered candidates to `launcher.judge` on each keystroke, re-orders results by the returned probabilities, and shows a green ↵ on the top row when intent is settled; mutating verbs like enable/disable MCP server or publish plugin marketplace require a second Enter to confirm.

— Exact endpoint and UI trigger, no threshold config shown dashboard@0.126.0 server@2.12.0

02 Slack workspace connection and directory sync NEW 80

Adds organization Slack workspace authorization — connect/disconnect/reauthorize multiple workspaces from Identity, with encrypted credentials and audited changes. Adds Slack directory sync that syncs on connection or on-demand, shows workspace sync history and member counts, and a searchable cross-workspace directory; connected workspaces sync every 30 minutes with automatic token refresh. Organization administrators can map Slack memberships to personnel via an inline picker and reassign or remove mappings, with exact-email matches applied automatically on sync; mapped Slack accounts then appear under Work identities on a person's Accounts & devices page.

— Concrete sync interval and matching rule, no API names server@2.12.0

03 Multi-host serving via GRAM_PLATFORM_HOSTS NEW 70

Serves the full product on extra first-party hosts listed in `GRAM_PLATFORM_HOSTS` (e.g. `ai.speakeasy.com`), with login and dashboard telemetry scoped to the originating host.

— Names exact env var and example host `server@2.12.0`

---

04 **Remote MCP server identity modes and setup** NEW 68

Configures Remote MCP server identity through a single atomic provider and client setup operation. Adds User, Agent, and No Identity modes for Remote MCP servers, including identity selection during creation, managed Authorization credentials, and editable Remote Identity Provider setup.

— Names identity modes, no config keys given `dashboard@0.126.0` `server@2.12.0`

---

05 **Identity-chaining client registration lifecycle controls** NEW 67

Adds management API and SDK operations to prepare, inspect, and unlink downstream identity-chaining client registrations with explicit grant evidence and generation checks. Active identity-chaining bindings are now protected during issuer lifecycle changes: unsafe issuer deletion and consolidation are blocked, with binding counts and unlinking guidance shown in the dashboard and admin migration review.

— Describes mechanism, no endpoint names given `server@2.12.0`

---

06 **MCP-scoped risk policies** NEW 65

Allows risk policies to target selected MCP servers, gateways, and tools while restricting those policies to tool traffic. The MCP scope picker in the risk policy editor is gated behind the `gram-mcp-scoped-policies` rollout flag.

— Names the rollout flag, lacks scoping mechanism detail `server@2.12.0` `dashboard@0.126.0`

---

07 **API key creation panel redesign** IMPROVED 65

Redesigns API key creation into a side pane (replacing a modal) with per-scope cards; removes the Chat scope from new key creation while preserving it on existing keys; sorts the project binding list alphabetically.

— Exact UI change and scope removal noted `dashboard@0.126.0`

THINNER COVERAGE BELOW

08 **New OpenRouter models in model pickers** NEW 58

Adds the latest OpenRouter models to the playground, chat, and assistant model pickers: Claude Opus 5.5, Claude Fable 5.1, GPT-6 Astra, GPT-6 Sol, GPT-6 Luna, Gemini 3.8 Flash, DeepSeek V4.1 Flash, Grok 4.7, Qwen3.8 Max, Qwen3.8 Flash, GLM-5.3, and Kimi K3.

— Lists every model, no rollout mechanism `dashboard@0.126.0`

09 **Read-only staff Admin MCP tools** NEW 57

Adds read-only staff Admin MCP tools for global issuers (list, detail, duplicate, migration preflights, convergence candidates), support matrix facts, organization onboarding configuration, project MCP server inventory, and bounded organization inference key state, spend history, and metered usage totals.

— Lists many tools but no invocation details `server@2.12.0`

10 **MCP Servers view in admin organization navigator** NEW 57

Adds an MCP Servers view to the admin organization navigator listing a project's MCP servers with URL, visibility, source, and creation date, with one-click URL copy.

— Exact fields shown, clear UI location `server@2.12.0`

11 **Consolidated identity-provider setup task** IMPROVED 52

Consolidates identity-provider setup on the setup board into one Set up identity provider task with three ordered steps: verify a domain, connect single sign-on, and sync the directory.

— Exact steps named, straightforward UI flow `server@2.12.0`

12 **Onboarding survey drives setup wizard** NEW 50

Adds a `submitOnboardingSurvey` endpoint that drives setup-wizard task selection from the onboarding survey result.

— Named endpoint, no detail on task mapping `server@2.12.0`

13 **OpenCode observability plugin and loaders** NEW 43

Generates an OpenCode observability plugin and OpenCode plugin loaders compatible with both OpenCode 1 and OpenCode 2.

— Names compatibility versions, no usage steps `server@2.12.0`

14 **Shadow AI scan toggle in fleet config** IMPROVED 42

Lets organization admins turn off the device agent's Shadow AI scan from the fleet configuration.

— Names the setting, lacks exact config key `server@2.12.0`

15 **Advertised authorization grant profiles in identity-provider models**

NEW 30

Exposes advertised authorization grant profiles in identity-provider API and SDK models.

— Bare description, no fields or examples named `server@2.12.0`

`server@2.12.0` NOTES ↗ LATEST STABLE

gram server 2.12.0 adds workload identity APIs, Slack directory sync, MCP-scoped risk policies, and directory role mappings.

└──▷ **GET THIS VERSION**

```
$ git clone --branch server@2.12.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout server@2.12.0
```

Adds the `workloadIdentities` management API with endpoints to read, register, and withdraw external issuers (`workload:read` / `workload:write`), admit and withdraw subjects, and audit every write under `workload-issuer` and `workload-admission` subjects with before/after snapshots.

Adds `allow_wildcard_admission` to `workload_issuers` and `match_kind` to `workload_identity_admissions` and `workload_agent_assignments`, enabling trailing-wildcard subject matching (e.g. `wimse://identity.anthropic.com/org/<ORG_ID>/agent/*`) under issuers that opt in.

Adds the `launcher.judge` management endpoint (`POST /rpc/launcher.judge`) that routes command-palette intent decisions to Jev via OpenRouter, returning probability distributions keyed by candidate IDs; organizations without an OpenRouter key receive `disabled: true`.

Adds the `admin.getSupportCoverage` endpoint, reporting per-surface evidence for session activity, policy enforcement, identity attribution, token usage, and shadow MCP exposure with explicit status for every (capability, surface) pair.

Adds `workload:read` and `workload:write` RBAC scopes gating workload identity trust-policy management; both are admin system-role defaults and are registered as agent-runtime scopes but deliberately not agent-runtime-safe.

+18 more

dashboard@0.126.0 NOTES ↗ STABLE

gram: Adds `workload:read` and `workload:write` permission scopes to gate view and change access on the new Workload Identities page, where administrators trust external issuers, admit subjects, and assign the inheriting agent policy in one action.

└──▷ **GET THIS VERSION**

```
$ git clone --branch dashboard@0.126.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout dashboard@0.126.0
```

Adds `launcher.judge` AI ranking to the command palette (⌘K): on each keystroke it sends typed text and prefiltered candidates to `launcher.judge`, re-orders results by Jev probability distributions, and shows a green ↵ on the top row when intent is settled; mutating verbs (enable/disable MCP server, publish plugin marketplace) require a second Enter.

Gates the MCP scope picker in the risk policy editor behind the `gram-mcp-scoped-policies` rollout flag.

Adds User, Agent, and No Identity modes for Remote MCP servers, including identity selection during creation, managed Authorization credentials, and editable Remote Identity Provider setup.

Manages directory role mappings on the IDP and SSO page: every directory group is listed with its own role picker and Create role… opens the role editor and maps the new role on return.

Adds the latest OpenRouter models to the playground, chat, and assistant model pickers: Claude Opus 5.5, Claude Fable 5.1, GPT-6 Astra, GPT-6 Sol, GPT-6 Luna, Gemini 3.8 Flash, DeepSeek V4.1 Flash, Grok 4.7, Qwen3.8 Max, Qwen3.8 Flash, GLM-5.3, and Kimi K3.

+1 more

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

HeyGen HyperFrames

Feedback ⓘ v0.8.82 NOTES ↗ RANK

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

> HyperFrames added a `findProjects` Core API, opt-in Parakeet CLI transcription, host-mountable Studio UI components, and gave Studio Server hosts the ability to override undo across collaborators.

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

01 **`findProjects` Core API for project enumeration** NEW — 60

Adds `findProjects` to the Core API to enumerate HyperFrames projects on a machine.

– Names the API method but no params or usage shown v0.8.82

THINNER COVERAGE BELOW

02 **Parakeet transcription in CLI** NEW — 55

Supports Parakeet transcription in the CLI on macOS, Linux, and Windows after an opt-in install.

– Platforms and opt-in noted but no install command given v0.8.82

03 **Studio Server host undo override** NEW — 52

Enables a Studio Server host to invoke Cmd+Z to undo the newest change regardless of which collaborator made it.

– Clear mechanism and scope, but no config or API named v0.8.81

04 **Host-mountable Studio UI components** NEW — 45

Lets a host app mount the timeline toolbar, undo buttons, meters, and clipboard in Studio.

– Components named but no mounting API or steps v0.8.82

05 **CLI telemetry tagged by launching app** IMPROVED — 30

Tags CLI telemetry and feedback with the app that launched the CLI.

– Behavior described only in general terms v0.8.82

06 **Portable agent bundles for Plugins** NEW — 25

Ships portable agent bundles and installation guides for Plugins.

— No details on bundle contents or install steps v0.8.82

v0.8.82 NOTES ↗ LATEST STABLE

HyperFrames v0.8.82 adds Parakeet transcription on CLI, a `findProjects` API, portable agent bundles, and host-mountable Studio UI components.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.82 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.82
```

Adds `findProjects` to the Core API to enumerate HyperFrames projects on a machine.

Supports Parakeet transcription in the CLI on macOS, Linux, and Windows after an opt-in install.

Lets a host app mount the timeline toolbar, undo buttons, meters, and clipboard in Studio.

Tags CLI telemetry and feedback with the app that launched the CLI.

Ships portable agent bundles and installation guides for Plugins.

v0.8.81 NOTES ↗ STABLE

HyperFrames v0.8.81 lets a Studio Server host trigger Cmd+Z undo for any collaborator's latest change.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.81 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.81
```

WAS THIS USEFUL? 👍 👎

Nebius AI Cloud

Feedback ⓘ September 21–27, 2026 (2026-09-28)

BLOG ↗ RANK

Nebius AI Cloud provides GPU compute, managed Kubernetes, distributed training infrastructure, and serverless AI workloads.

> Nebius AI Cloud shipped automatic job restarts and spot pricing for compute, encrypted CSI-backed disks, cost allocation labels, several Managed Kubernetes improvements, three newly public regions, and an updated Tunnels agent.

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

01 **Encrypted CSI disks in Kubernetes** NEW 85

Enables encryption for `NETWORK_SSD_NON_REPLICATED` and `NETWORK_SSD_IO_M3` disks provisioned over CSI by setting `encrypted: "true"` in a custom Kubernetes `StorageClass` (provisioner `disk.csi.nebius.ai`).

Provision an encrypted non-replicated SSD disk in Kubernetes via CSI to meet data-at-rest requirements without manual key management.

yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: encrypted-nonreplicated
provisioner: disk.csi.nebius.ai
parameters:
  type: NETWORK_SSD_NON_REPLICATED
  encrypted: "true"
```

Provision an encrypted high-performance disk via CSI in Kubernetes for workloads with data-at-rest requirements.

yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: encrypted-ssd
provisioner: disk.csi.nebius.ai
parameters:
  type: NETWORK_SSD_NON_REPLICATED
  encrypted: "true"
```

— Runnable StorageClass YAML with exact parameters provided

September 21–27, 2026 (2026-09-28)

02 **Automatic restart for Serverless AI jobs** NEW 75

Adds a `--restart-policy on-failure` flag for Serverless AI jobs that automatically restarts containers which exit with a non-zero code.

— Names exact flag but no example of full command September 21–27, 2026 (2026-09-28)

03 **Spot pricing for preemptible VMs and GPUs** NEW 75

Adds spot pricing for preemptible VMs and GPU platforms, with a per-resource maximum price cap or a Follow spot price policy; spot prices update every 15 minutes.

— Mechanism and cadence given but no config/flag shown September 21–27, 2026 (2026-09-28)

04 **Managed Kubernetes logging, version and drain updates** NEW 70

Adds journald log collection from Managed Service for Kubernetes node groups, viewable in Observability Logs (disabled by default); adds Kubernetes 1.36 support as the recommended version for new clusters; and sets a default drain timeout of 10 minutes for node groups, after which nodes are force-drained and deleted.

— Concrete defaults and version given, no flags to invoke

September 21–27, 2026 (2026-09-28)

05 **Cost allocation labels for Compute billing** NEW 65

Adds cost allocation labels to group Compute costs by service, application, or team, filterable on the Billing → Usage page and visible in the Tags column of exported usage data.

— UI path and column named but no config key September 21–27, 2026 (2026-09-28)

THINNER COVERAGE BELOW

06 **Nebius Tunnels agent v1.2.0** IMPROVED 55

Updates the Nebius Tunnels agent to v1.2.0, adding support for connections to local services, more idle connection reuse, and reduced CPU usage per forwarded request.

— Names version and improvements but no config/command September 21–27, 2026 (2026-09-28)

07 **New public regions** NEW 40

Makes the `us-north1`, `eu-south1`, and `eu-west2` regions publicly available to all Nebius users.

— Bare announcement, no usage detail beyond region names

September 21–27, 2026 (2026-09-28)

September 21–27, 2026 (2026-09-28) BLOG ↗ LATEST STABLE

Nebius AI Cloud adds spot pricing for preemptible VMs, journald log collection for Kubernetes, and automatic job restarts via `--restart-policy`.

└──▷ **USE IT**

Provision an encrypted high-performance disk via CSI in Kubernetes for workloads with data-at-rest requirements.

yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: encrypted-ssd
provisioner: disk.csi.nebius.ai
parameters:
  type: NETWORK_SSD_NON_REPLICATED
  encrypted: "true"
```

Adds `--restart-policy on-failure` flag for Serverless AI jobs to automatically restart containers that exit with a non-zero code.

Enables encryption for `NETWORK_SSD_NON_REPLICATED` and `NETWORK_SSD_IO_M3` disks provisioned over CSI by setting `encrypted: "true"` in a custom storage class.

Adds spot pricing for preemptible VMs and GPU platforms, with a per-resource maximum price cap or a `Follow spot price` policy; spot price updates every 15 minutes.

Adds cost allocation labels to group Compute costs by service, application, or team, filterable on the Billing → Usage page and visible in the `Tags` column of exported usage data.

Adds journald log collection from Managed Service for Kubernetes node groups, viewable in Observability Logs (disabled by default).

+4 more

WAS THIS USEFUL? 👍 👎

BerriAI LiteLLM

Feedback ⓘ v1.103.0 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 v1.103.0 adds config-level controls for file uploads, JWT claim mapping, and access logging, alongside S3-backed managed file operations for Bedrock and a non-blocking logging mode for Model Armor guardrails.

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

01 File upload extension allowlisting NEW 75

Adds `general_settings.allowed_file_extensions` config key to restrict which file types are accepted on `/v1/files` uploads, for example restricting uploads to only `pdf` and `txt` to keep unexpected file types from reaching Bedrock or OpenAI backends.

Restrict file uploads to only PDF and plain-text files to prevent unexpected file types reaching your Bedrock or OpenAI backends.

yaml

```yaml
general_settings:
  allowed_file_extensions:
    - pdf
    - txt
```

— exact config key and working YAML example given v1.103.0

02 Access-log path suppression via env var NEW 75

Honors the `LITELLM_DISABLE_ACCESS_LOG_PATHS` environment variable to suppress noisy uvicorn access-log lines for specified paths, such as silencing `/health` and `/metrics` polling in production.

Silence health-check and metrics poll endpoints from flooding uvicorn access logs in production.

env

```env
LITELLM_DISABLE_ACCESS_LOG_PATHS=/health,/metrics
```

— named env var with concrete usage example v1.103.0

03 Logging-only mode for Model Armor guardrails NEW 60

Adds `logging_only` mode for Model Armor guardrails, which scans completed streams post-delivery without blocking responses.

— explains behaviour but no config snippet given v1.103.0

THINNER COVERAGE BELOW

04 **Per-issuer JWT claim field for virtual keys** NEW 55

Adds `virtual_key_claim_field` per-issuer configuration for JWT authentication, allowing each issuer to name its own claim field used for virtual key lookup.

— names config key but no usage example shown v1.103.0

05 **Provider-scoped model_info backfill** NEW 45

Adds `fill_missing_for_providers` provider-scoped backfill in `model_info` using fallback generalization rules.

— names config key but mechanism underspecified v1.103.0

06 **S3-backed managed files for Bedrock** NEW 40

Supports file delete and list operations for S3-backed managed files on Bedrock.

— names the operations but no mechanism or config detail v1.103.0

07 **Prompt-cache cost prediction across deployments** NEW 25

Adds prompt-cache cost prediction across deployments in the proxy.

— bare description, no mechanism or usage v1.103.0

08 **Friendli price registry auto-sync** NEW 25

Auto-syncs Friendli model metadata into the price registry.

— bare description, no mechanism or usage v1.103.0

v1.103.0 NOTES ↗ LATEST STABLE

LiteLLM v1.103.0 adds S3-backed file management for Bedrock, per-issuer JWT claim fields, file-upload extension allowlisting, access-log suppression, model_armor logging-only

└──▷ GET THIS VERSION

```
$ git clone --branch v1.103.0 https://github.com/BerriAI/litellm.git
# already have the repo? check out this version:
$ git checkout v1.103.0
```

└──▷ **USE IT**

Restrict file uploads to only PDF and plain-text files to prevent unexpected file types reaching your Bedrock or OpenAI backends.

litellm_config.yaml

```
general_settings:
  allowed_file_extensions:
    - pdf
    - txt
```

Silence health-check and metrics poll endpoints from flooding uvicorn access logs in production.

.env

```
LITELLM_DISABLE_ACCESS_LOG_PATHS=/health,/metrics
```

Adds `general_settings.allowed_file_extensions` config key to restrict which file types are accepted on `/v1/files` uploads.

Honors the `LITELLM_DISABLE_ACCESS_LOG_PATHS` environment variable to suppress noisy uvicorn access-log lines for specified paths.

Adds `virtual_key_claim_field` per-issuer configuration for JWT authentication, allowing each issuer to name its own claim field for virtual key lookup.

Supports file delete and list operations for S3-backed managed files on Bedrock.

Adds `logging_only` mode for Model Armor guardrails, which scans completed streams post-delivery without blocking responses.

+3 more

WAS THIS USEFUL? 👍 👎

Eigen Labs Darkbloom

Feedback ⓘ v0.9.11 NOTES ↗

CODE ↗ RANK

Private Inference Network on Idle Macs

> Darkbloom improved Qwen prefix-cache efficiency with demand-gated SSD checkpoint writes and longer cache lifetimes, while raising the minimum provider version required to stay in the routing pool.

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

01 **Demand-gated SSD checkpoint writes** NEW 78

Darkbloom now only writes a complete SSD checkpoint when `cache_repeated_prefix_tokens` is at or above the 1,024-token floor, or a prior local sighting already exists. In practice the first request for a prefix is served cold, the second triggers the write, and the third can hit the cache — improving prefix-cache hit rates for Qwen models.

— Names exact config key and threshold with full before/after behaviour. v0.9.11

02 **Raised minimum provider version for routing** BREAKING 77

The routing floor for providers is raised to 0.9.5 via `EIGENINFERENCE_MIN_PROVIDER_VERSION=0.9.5` set in `/etc/d-inference/env`; the 25 providers currently below 0.9.5 (2.2% of the fleet) will stop receiving traffic once this is applied.

— Exact env var, config file path and impact figures given. v0.9.11

THINNER COVERAGE BELOW

03 **Longer SSD prefix-cache TTL for providers** IMPROVED 41

SSD prefix-cache TTL for provider nodes is raised from 15 to 30 minutes, as part of smarter routing changes aimed at improving prefix-cache hit rates for Qwen models.

— States exact before/after values but no mechanism beyond that. v0.9.11

└──▷ **BREAKING ON UPGRADE**

! Routing floor for providers is raised to 0.9.5 via `EIGENINFERENCE_MIN_PROVIDER_VERSION=0.9.5` in `/etc/d-inference/env`; the 25 providers (2.2% of fleet) currently below 0.9.5 will stop receiving traffic once this is applied.

v0.9.11 NOTES ↗ CODE ↗ LATEST STABLE

Darkbloom v0.9.11 improves prefix-cache hit rates for Qwen models with smarter routing, longer TTLs, and demand-gated writes.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.9.11 https://github.com/Layr-Labs/d-inference.git
# already have the repo? check out this version:
$ git checkout v0.9.11
```

Introduces demand-gated checkpoint writes: only writes a complete SSD checkpoint when `cache_repeated_prefix_tokens` is at or above the 1,024-token floor, or a prior local sighting exists — first request is served cold, second is written, third can hit.

Raises SSD prefix-cache TTL from 15 to 30 minutes for provider nodes.

└──▷ **BREAKING ON UPGRADE**

! Routing floor for providers is raised to 0.9.5 via `EIGENINFERENCE_MIN_PROVIDER_VERSION=0.9.5` in `/etc/d-inference/env`; the 25 providers (2.2% of fleet) currently below 0.9.5 will stop receiving traffic once this is applied.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

InternLM LMDeploy

Feedback ⓘ v0.18.0 NOTES ↗ RANK

LMDeploy compresses, deploys, and serves language and vision-language models with its TurboMind engine, offering quantization and an OpenAI-compatible API.

> LMDeploy v0.18.0 extends its serving API with input/output logprobs, restructures MoE FFN execution for better multi-node throughput, and broadens quantized inference with new TurboMind backends, attention kernels, and KV-connector support for Qwen3.5, GPT-OSS, and Mooncake.

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

01 **MoE FFN sharding and shared-expert placement** IMPROVED

60

Moves the shared expert into the MoE FFN layer and shards the dense FFN node-locally in MoE models, improving multi-node MoE throughput.

— Describes before/after mechanism and stated benefit. v0.18.0

THINNER COVERAGE BELOW

02 **Structured output via xgrammar structural_tag and Harmony parser for GPT-OSS**

NEW

55

Adds `xgrammar` `structural_tag` grammar combined with the Harmony parser for structured output generation in GPT-OSS models.

— Names grammar, parser, and target model but no usage steps. v0.18.0

03 **Expanded SM90 quantized GEMMs and unified TurboMind linear execution**

IMPROVED

55

Expands SM90 quantized GEMMs and unifies TurboMind linear execution, broadening quantized inference coverage on Hopper GPUs.

— Explains scope and target hardware but no config specifics. v0.18.0

04 **TurboMind W4A16 backend for AWQ linear (prototype)** NEW

55

Adds a TurboMind W4A16 backend for AWQ linear in the PyTorch engine (prototype).

— Names precision, quantization scheme and engine, marked prototype. v0.18.0

05 **dflash attention for Qwen3.5** NEW

45

Adds `dflash` attention support for Qwen3.5 models.

— Names the backend but not how to enable it. v0.18.0

06 **Input and output logprobs in serving API** NEW 40

Adds support for returning both output and input logprobs through the serving API, extending token probability introspection for LMDeploy-served models.

— States capability but no endpoint or param named. v0.18.0

07 **Mooncake store support for MTP in KV connector** NEW 40

Adds Mooncake store support for multi-token prediction (MTP) in the KV connector.

— Names the integration but not configuration details. v0.18.0

08 **Piecewise CUDA graph prefill for PyTorch engine** NEW 30

Adds piecewise CUDA graph prefill for the PyTorch engine.

— Bare one-line description, no mechanism given. v0.18.0

09 **XTuner TileLang sparse MLA backend** NEW 30

Adds XTuner TileLang sparse MLA backend.

— Named but undescribed beyond the title. v0.18.0

10 **Symmetric-memory LM-head all-gather** NEW 30

Adds symmetric-memory LM-head all-gather.

— Named optimization but no explanation of benefit. v0.18.0

11 **Request-only cache usage metric** NEW 25

Adds a request-only cache usage metric.

— No detail on what the metric measures or exposes. v0.18.0

12 **Checkpoint-engine weight update support** NEW 25

Adds checkpoint-engine weight update support.

— No mechanism or usage described. v0.18.0

v0.18.0 NOTES ↗ LATEST STABLE

LMDeploy v0.18.0 adds input logprobs, GLM-5.2B Ascend support, Qwen3.5 dflash, MoE FFN sharding, and Mooncake MTP for KV connector.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.18.0
https://github.com/InternLM/lmdeploy.git
# already have the repo? check out this version:
$ git checkout v0.18.0
```

Adds output and input logprobs support via the serving API.

Adds `dflash` attention support for Qwen3.5 models.

Adds piecewise CUDA graph prefill for the PyTorch engine.

Adds XTuner TileLang sparse MLA backend.

Adds request-only cache usage metric.

+7 more

WAS THIS USEFUL?  

vMLX

Feedback ⓘ v1.6.69 NOTES ↗ RANK

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

> vMLX 1.6.69 introduces an opt-in second-window confirmation step for Native MTP, expands internal diagnostics to separate forward-call, wait, and CPU thread timings, and clarifies how input-rate figures are shown in chat metrics.

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

01 **Expanded forward-call and thread diagnostics** IMPROVED

55

Diagnostics now distinguish forward-call wall time, completion wait, and calling-thread CPU usage, with bounded verifier-row host profiling and isolated thread accounting.

— Describes what changed but no command or key to invoke it v1.6.69

02 **Second-window confirmation for Native MTP** NEW

45

Adds an opt-in second-window confirmation for marginal D1 losses in Native MTP; the feature is disabled by default.

— Names mechanism (D1 loss, second window) but no config flag given v1.6.69

03 **Clarified input-rate display in chat metrics** IMPROVED

40

Hides the derived input-rate figure on cache hits, and explicitly labels cold-request input rate as input divided by TTFT in chat metrics.

— Explains the display change with no UI path or flag v1.6.69

v1.6.69 NOTES ↗ LATEST STABLE

vMLX 1.6.69 adds opt-in Native MTP second-window confirmation and expanded diagnostics for forward-call, wait, and CPU thread profiling.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.6.69 https://github.com/jjang-ai/vmlx.git
# already have the repo? check out this version:
$ git checkout v1.6.69
```

Adds opt-in second-window confirmation for marginal D1 losses in Native MTP, disabled by default.

> Expands diagnostics to distinguish forward-call wall time, completion wait, and calling-thread CPU, with bounded verifier-row host profiling and isolated thread accounting.
>
> Hides the derived input-rate figure on cache hits and labels cold-request input divided by TTFT explicitly in chat metrics.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Volcengine OpenViking

Feedback ⓘ v0.4.22 NOTES ↗

RANK

OpenViking is a context database that manages agent memory, knowledge retrieval, and skills.

> OpenViking v0.4.22 ships a two-level runtime configuration API for restart-free account overrides, adds ACL support at resource-creation time, overhauls Skill package indexing and catalog injection, and introduces new backends and integrations alongside a long list of breaking changes to sessions, scoring, and tool naming.

└──▷ WHAT SHIPPED · 22 FEATURES most completely described first ? what's the number?

01 Two-level runtime configuration API NEW

95

Introduces Cluster-level and Account-level runtime configuration via `GET/PATCH /api/v1/admin/configuration` and `GET/PATCH /api/v1/admin/accounts/{account_id}/configuration`, letting VLM, Query Planner, Embedding, remote VectorDB, and Feishu credentials be changed per account without editing `ov.conf` or restarting. Adds the deployment-level `server.user_config_defaults.auto_commit_policy` config key, and deprecates `GET/PATCH /api/v1/admin/accounts/{account_id}/settings` and `GET/PUT /api/v1/admin/agent-evolution` in favor of the new API; during rolling upgrades with shared storage, avoid writing account fields unrecognized by the old version since older replicas may fail to parse `/local/{account_id}/_system/setting.json`.

Override an account's VLM or ACL settings at runtime without restarting the service or editing ov.conf.

```
$ curl -X PATCH
"$OV_URL/api/v1/admin/accounts/acme/configuration" \
  -H "X-API-Key: $ROOT_KEY" \
  -H 'Content-Type: application/json' \
  -d '{"acl": {"enabled": true}}'
```

— Endpoints, config key, and migration path fully specified. v0.4.22

02 ACL support at resource creation BREAKING

95

Adds `acl` parameter support to `add_resource`, `mkdir`, and `write` API calls (and the CLI's `--acl` flag) so access control can be set at creation time. With ACL enabled, the shared root now defaults to inherited `user:* = manage` instead of granting direct

manage to the creator, so directories requiring restricted access must explicitly set `acl_mode: restricted`; passing `acl` on create/import now requires manage permission, and write-only callers receive `403`.

Create a restricted project directory with explicit ACL entries at mkdir time, granting alice manage and the dev group read access.

```
$ curl -X POST "$OV_URL/api/v1/fs/mkdir" \
  -H "X-API-Key: $KEY" \
  -H 'Content-Type: application/json' \
  -d '{"uri":"viking://resources/project-a","acl":
{"acl_mode":"restricted","entries":
[{"principal":"user:alice","level":"manage"},
{"principal":"group:dev","level":"read"}]}}'
```

— Endpoint calls, flag, and permission changes named exactly. v0.4.22

03 **Skill package indexing and catalog** BREAKING 95

Adds the `openviking-skills` skill and injects a Skill catalog at session start via memory plugins. Whole Skill packages are now indexed, so `skills/find` and search(mode="context") return one best hit per Skill, with `root_uri` identifying the package root while `uri` may now point at L0, L1, or a file inside the package instead of the package root. Existing Skills are not reprocessed on upgrade; run `ov reindex viking://agent/skills --mode semantic_and_vectors` to rebuild.

Rebuild Skill package-level summaries and indexes after upgrading so that skills/find returns the best per-Skill hit.

```
$ ov reindex viking://agent/skills --mode semantic_and_vectors
```

— Mechanism, before/after, and rebuild command all given. v0.4.22

04 **Session.used() API removed** BREAKING 75

`POST /api/v1/sessions/{session_id}/used`, Python Session.used(), the Studio `/session used` command, and the matching generated client method are removed; calls now return `404`. `contexts_used` and `skills_used` are also removed from session metrics and storage stats.

— Endpoint, methods, and removed fields all named. v0.4.22

05 **Cosine score range remapped** BREAKING 75

Local vector engine cosine scores are remapped from `[-1,1]` to `[0,1]` via `(cos + 1) / 2`; ranking is unchanged but numeric values differ (e.g. raw 0 → 0.5, -0.6 → 0.2), so callers using `score_threshold` or score filters on cosine indexes must recalibrate thresholds.

— Formula and before/after values fully specified. v0.4.22

06 **Session commit failure reporting** BREAKING 70

Session commit tasks and archives are now marked `failed` (with `failure_kind=parse_error` or `failure_kind=empty_response`) when memory extraction output is unparseable after all retries; previously they reported success with zero memories, so callers that alert or retry on task status must handle the new `failed` outcome.

— Failure states and behavior change clearly specified. v0.4.22

07 **Directory parser file limit default changed** BREAKING 70

`parsers.directory.max_files` default changes from `1000` to `null` (unlimited); set it explicitly to `1000` in `ov.conf` to restore the previous limit.

— Config key, defaults, and fix all named. v0.4.22

08 **VikingBot sandbox now managed by default** BREAKING 70

VikingBot: `bot.sandbox.backend=opensandbox` now defaults to `managed=true`, meaning Gateway/ `--with-bot` manages the Docker OpenSandbox; existing external OpenSandbox services must explicitly set `bot.sandbox.backends.opensandbox.managed=false`. Files in the old `bot/workspace/shared` are not migrated to the new working directory.

— Default change and migration keys specified. v0.4.22

09 **MCP tools renamed to openviking_ prefix** BREAKING 65

The pi extension 0.4.0 replaces 7 handwritten `viking_*` tools with server-side MCP tools registered as `openviking_<tool>`; Root API keys can no longer access `/mcp` and must be replaced with user or admin keys.

— Naming scheme and access change specified. v0.4.22

10 **MCP tool and prefix updates** NEW 60

MCP surface gains an `add_skill` tool; MCP `list` now defaults to the `viking://` prefix and `edit` reports CRLF/LF mismatches.

— Names tool and defaults but no usage example. v0.4.22

THINNER COVERAGE BELOW

11 **Studio account search and permission UI** NEW 50

Studio gains a case-insensitive `?query=` account search on the account list, Mermaid preview, resource permission management, and user group management.

— Lists concrete UI additions but no mechanism. v0.4.22

12 **New agent/editor integrations** NEW 50

Adds a Kimi Code CLI memory plugin, updates the OpenCode plugin to support OpenCode v2, and gives Hermes gateway memory presets and sender attribution.

— Names three integrations without mechanism detail. v0.4.22

13 **OpenClaw peer-role person deprecated** DEPRECATED 50

OpenClaw: `--peer-role person`, `OPENVIKING_PEER_ROLE=person`, and interactive input `person` now return an error; use `sender` instead. Existing `peer_role: "person"` in config continues to be treated as `sender`.

— Flags and replacement behavior clearly specified. v0.4.22

14 **Ripgrep removed from Docker image** BREAKING 45

The Docker image no longer ships ripgrep, and RAGFS no longer probes or invokes an external `rg` binary.

— Names removed dependency and its effect. v0.4.22

15 **Watch task reliability improvements** IMPROVED 40

Watch tasks add support for externally managed OAuth tokens, and Git Watch now auto-deactivates on authentication failure.

— Names two improvements without deep mechanism. v0.4.22

16 **Bulk tag clearing in tags API** NEW 35

Adds `tag_mode="clear"` to the tags API for bulk tag clearing.

— Names endpoint and mode value; no example given. v0.4.22

17 **JSON output for observer API** NEW 35

Adds a `?format=json` query parameter to the observer API.

— Names query parameter but lacks usage context. v0.4.22

18 **openGauss DataVec vector backend** NEW 35

Adds an openGauss DataVec vector backend.

— Just names the backend, no detail. v0.4.22

19 **Feishu mindnote parsing** NEW — 30

Adds support for Feishu mindnote parsing.

— Bare mention of new parsing support. v0.4.22

---

20 **Unicode-safe storage URI normalization** IMPROVED — 30

Storage URI normalization now handles Unicode and space-containing paths.

— States behavior but no mechanism or example. v0.4.22

---

21 **Jev rerank provider** NEW — 25

Adds a Jev rerank provider.

— Just names the provider, no detail. v0.4.22

---

22 **Context lines in grep** IMPROVED — 25

Grep gains support for context lines around matches.

— Bare feature name, no detail. v0.4.22

└──▷ **BREAKING ON UPGRADE**

! `POST /api/v1/sessions/{session_id}/used`, Python Session.used(), the Studio `/session used` command, and the matching generated client method are removed; calls return 404. `contexts_used` and `skills_used` are removed from session metrics and storage stats.

! Session commit tasks and archives are now marked `failed` (with `failure_kind=parse_error` or `failure_kind=empty_response`) when memory extraction output is unparseable after all retries; previously they reported success with zero memories. Callers that alert or retry on task status must handle the new `failed` outcome.

! `GET/PATCH /api/v1/admin/accounts/{account_id}/settings` and `GET/PUT /api/v1/admin/agent-evolution` are deprecated; migrate to the new configuration API. During a rolling upgrade with shared storage, do not write account fields unrecognized by the old version, as older replicas may fail to parse `/local/{account_id}/_system/setting.json`.

! Local vector engine cosine scores are remapped from `[-1,1]` to `[0,1]` via `(cos + 1) / 2`; ranking is unchanged but numeric values differ (e.g. raw 0 → 0.5, -0.6 → 0.2). Callers using `score_threshold` or score filters on cosine indexes must recalibrate thresholds.

! `skills/find` now returns one best hit per Skill and `uri` may point at L0, L1, or a file inside the package instead of the package root; code that used `uri` as the package root must switch to `root_uri`. Existing Skills are not reprocessed on upgrade; run `ov reindex viking://agent/skills --mode semantic_and_vectors` to rebuild.

! `parsers.directory.max_files` default changed from `1000` to `null` (unlimited). Set it explicitly to `1000` in `ov.conf` to restore the previous limit.

! With ACL enabled, the shared root now defaults to inherited `user:* = manage` instead of granting direct manage to the creator; directories requiring restricted access must set `acl_mode: restricted` explicitly. Passing `acl` on create/import now requires manage permission; write-only callers receive 403.

! The pi extension 0.4.0 replaces 7 handwritten `viking_*` tools with server-side MCP tools registered as `openviking_<tool>`; Root API keys cannot access `/mcp` and must be replaced with user or admin keys.

! OpenClaw: `--peer-role person`, `OPENVIKING_PEER_ROLE=person`, and interactive input `person` now return an error; use `sender` instead. Existing `peer_role: "person"` in config continues to be treated as `sender`.

! VikingBot: `bot.sandbox.backend=opensandbox` now defaults to `managed=true`, meaning Gateway/ `--with-bot` manages Docker OpenSandbox. Existing external OpenSandbox services must set `bot.sandbox.backends.opensandbox.managed=false` explicitly. Files in the old `bot/workspace/shared` are not migrated to the new working directory.

v0.4.22 NOTES ↗ LATEST STABLE

OpenViking v0.4.22 adds two-level runtime config API, openGauss DataVec backend, Jev rerank, Kimi Code CLI plugin, and Feishu mindnote parsing.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.4.22
https://github.com/volcengine/OpenViking.git
# already have the repo? check out this version:
$ git checkout v0.4.22
```

└──▷ **TRY IT**

Override an account's VLM or ACL settings at runtime without restarting the service or editing ov.conf.

```
$ curl -X PATCH "$OV_URL/api/v1/admin/accounts/acme/configuration" \
  -H "X-API-Key: $ROOT_KEY" \
  -H 'Content-Type: application/json' \
  -d '{"acl": {"enabled": true}}'
```

Create a restricted project directory with explicit ACL entries at mkdir time, granting alice manage and the dev group read access.

```
$ curl -X POST "$OV_URL/api/v1/fs/mkdir" \
  -H "X-API-Key: $KEY" \
  -H 'Content-Type: application/json' \
  -d '{"uri":"viking://resources/project-a","acl":
{"acl_mode":"restricted","entries":
[{"principal":"user:alice","level":"manage"},
{"principal":"group:dev","level":"read"}]}}'
```

Rebuild Skill package-level summaries and indexes after upgrading so that skills/find returns the best per-Skill hit.

```
$ ov reindex viking://agent/skills --mode semantic_and_vectors
```

Adds Cluster-level and Account-level runtime configuration via `GET/PATCH /api/v1/admin/configuration` and `GET/PATCH /api/v1/admin/accounts/{account_id}/configuration`, allowing VLM, Query Planner, Embedding, remote VectorDB, and Feishu credentials to be changed per account without editing `ov.conf` or restarting.

Adds deployment-level `server.user_config_defaults.auto_commit_policy` configuration key.

Adds `acl` parameter support to `add_resource`, `mkdir`, and `write` API calls so access control can be set at creation time; CLI exposes the same as `--acl`.

Adds `tag_mode="clear"` to the tags API for bulk tag clearing.

Adds `?format=json` query parameter to the observer API.

+18 more

└─▷ **BREAKING ON UPGRADE**

! `POST /api/v1/sessions/{session_id}/used`, Python Session.used(), the Studio `/session used` command, and the matching generated client method are removed; calls return 404. `contexts_used` and `skills_used` are removed from session metrics and storage stats.

! Session commit tasks and archives are now marked `failed` (with `failure_kind=parse_error` or `failure_kind=empty_response`) when memory extraction output is unparseable after all retries; previously they reported success with zero memories. Callers that alert or retry on task status must handle the new `failed` outcome.

! `GET/PATCH /api/v1/admin/accounts/{account_id}/settings` and `GET/PUT /api/v1/admin/agent-evolution` are deprecated; migrate to the new configuration API. During a rolling upgrade with shared storage, do not write account fields unrecognized by the old version, as older replicas may fail to parse `/local/{account_id}/_system/setting.json`.

! Local vector engine cosine scores are remapped from `[-1,1]` to `[0,1]` via `(cos + 1) / 2`; ranking is unchanged but numeric values differ (e.g. raw 0 → 0.5, -0.6 → 0.2). Callers using `score_threshold` or score filters on cosine indexes must recalibrate thresholds.

! `skills/find` now returns one best hit per Skill and `uri` may point at L0, L1, or a file inside the package instead of the package root; code that used `uri` as the package root must switch to `root_uri`. Existing Skills are not reprocessed on

upgrade; run `ov reindex viking://agent/skills --mode semantic_and_vectors` to rebuild.

- ! `parsers.directory.max_files` default changed from `1000` to `null` (unlimited). Set it explicitly to `1000` in `ov.conf` to restore the previous limit.
- ! With ACL enabled, the shared root now defaults to inherited `user:* = manage` instead of granting direct manage to the creator; directories requiring restricted access must set `acl_mode: restricted` explicitly. Passing `acl` on create/import now requires manage permission; write-only callers receive 403.
- ! The pi extension 0.4.0 replaces 7 handwritten `viking_*` tools with server-side MCP tools registered as `openviking_<tool>`; Root API keys cannot access `/mcp` and must be replaced with user or admin keys.
- ! OpenClaw: `--peer-role person`, `OPENVIKING_PEER_ROLE=person`, and interactive input `person` now return an error; use `sender` instead. Existing `peer_role: "person"` in config continues to be treated as `sender`.
- ! VikingBot: `bot.sandbox.backend=opensandbox` now defaults to `managed=true`, meaning Gateway/ `--with-bot` manages Docker OpenSandbox. Existing external OpenSandbox services must set `bot.sandbox.backends.opensandbox.managed=false` explicitly. Files in the old `bot/workspace/shared` are not migrated to the new working directory.

WAS THIS USEFUL? 👍 👎

okf-memory OKF Agent Memory

Feedback ⓘ v0.4.4

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's v0.4.4 release adds AST-based frontmatter filtering to search, a stale-review horizon check for validate, full type/status/tag parity across CLI and MCP tools, and refactors internals into reusable Go packages.

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

01 Type/status/tags parity across CLI and MCP IMPROVED

93

Adds `--type`, `--status`, and `--tags` flags to `okf update` with explicit replacement semantics, where passing `--tags ""` clears all concept tags; adds a `--status` flag to `okf create`, defaulting to `stable`, with upfront validation enforcing `draft`, `stable`, or `deprecated`. Mirrors this on the MCP side: the `okf_update` MCP tool exposes `type`, `status`, `tags`, `title`, `description`, and `body` fields (`tags: []` clears tags), and the `okf_create` MCP tool exposes optional `status` and `tags` fields with a 50-character per-tag limit (`maxLength: 50`) enforced across all schemas.

Promote a draft decision to stable and replace its tags in one operation, replacing whatever tags existed before.

```
$ ./bin/okf update decisions/auth-flow knowledge --status stable
--type Decision --tags "oauth2,pkce"
```

— Enumerates every flag, field and constraint verbatim. v0.4.4

02 AST filter queries in okf search NEW

88

Adds `--filter "<expr>"` to `okf search` for AST-based frontmatter queries supporting `=`, `!=`, `<`, `<=`, `>`, `>=`, `=~` (regex), and `contains` operators, e.g. filtering results to concepts whose `status` isn't deprecated and whose `tags` contain a given value.

Slice search results to only concepts tagged `auth` whose status is not deprecated — useful for finding active, relevant memory entries before modifying code.

```
$ ./bin/okf search "architecture layers" knowledge --filter
"status != deprecated" --filter "tags contains auth"
```

— Runnable command with named operators and example. v0.4.4

03 **--stale-within review horizon in validate** NEW 81

Adds `--stale-within <duration>` to `okf validate` to project forward review horizons against `stale_after` dates, letting CI surface concepts overdue for review within a given window (e.g. 30 days) before memory rot misleads agents.

Surface stale concepts in CI that are overdue for review within the next 30 days, catching memory rot before it misleads agents.

```
$ ./bin/okf validate knowledge --drift --stale-within 30d
```

— Named flag with concrete CI use case and example. v0.4.4

04 **Reusable pkg/okf and pkg/sync packages** NEW 65

Decouples the embedded stdio MCP server into a reusable `pkg/okf` package and Hub vault/sync operations into `pkg/sync`, introducing `okf.SaveOptions` and `HubOp` structures for deterministic programmatic bundle interactions.

— Names new packages and types but gives no usage example. v0.4.4

v0.4.4 NOTES ↗ LATEST STABLE

okf v0.4.4 adds `--filter` AST queries and `--stale-within` to validate, plus full `--type` / `--status` / `--tags` parity across CLI and MCP.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.4.4 https://github.com/okf-memory/okf-agent-
memory.git
# already have the repo? check out this version:
$ git checkout v0.4.4
```

└──▷ TRY IT

Slice search results to only concepts tagged `auth` whose status is not deprecated — useful for finding active, relevant memory entries before modifying code.

```
$ ./bin/okf search "architecture layers" knowledge --filter "status
!= deprecated" --filter "tags contains auth"
```

Surface stale concepts in CI that are overdue for review within the next 30 days, catching memory rot before it misleads agents.

```
$ ./bin/okf validate knowledge --drift --stale-within 30d
```

Promote a draft decision to stable and replace its tags in one operation, replacing whatever tags existed before.

```
$ ./bin/okf update decisions/auth-flow knowledge --status stable --
type Decision --tags "oauth2,pkce"
```

Adds `--filter "<expr>"` to `okf search` for AST-based frontmatter queries supporting `=`, `!=`, `<`, `<=`, `>`, `>=`, `=~` (regex), and `contains` operators.

Adds `--stale-within <duration>` to `okf validate` to project forward review horizons against `stale_after` dates.

Adds `--type`, `--status`, and `--tags` flags to `okf update` with explicit replacement semantics; passing `--tags ""` clears all concept tags.

Adds `--status` flag to `okf create`, defaulting to `stable`, with upfront validation enforcing `draft`, `stable`, or `deprecated` values.

Exposes `type`, `status`, `tags`, `title`, `description`, and `body` fields on the `okf_update` MCP tool; passing `tags: []` clears tags.

+3 more

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20260928 NOTES ↗

RANK

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

> LangSmith's window centers on LLM Gateway improvements (model pricing metadata, fallback timeout handling, auth context propagation) alongside expanded OAuth support for agents and shared workspace credentials, plus a breaking change tightening dataset export permissions.

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

01 LLM Gateway model config, fallback and routing IMPROVED

75

LLM Gateway adds a `ls_provider` field on OpenAI-compatible model configurations to populate trace metadata and match custom models to provider-specific pricing. Fallback chains can now advance to the next model when a candidate does not return response headers within a configured timeout, while active response streams continue after headers arrive. Gateway requests authenticated with a session or OAuth bearer token retain authentication and workspace context across internal routing, including model fallbacks.

Route custom model traffic through the LLM Gateway and attribute traces to the correct provider pricing by setting `ls_provider` in the model configuration.

json

```
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-gateway.example.com/v1",
  "ls_provider": "openai"
}
```

— Names field, config shape and mechanism; no full endpoint given snapshot-20260928

02 License expiration warning window for self-hosted NEW

70

Self-hosted operators can configure how many days before license expiration the warning banner appears via the `LICENSE_EXPIRATION_WARNING_DAYS` environment variable, default 7.

Give self-hosted teams earlier notice before a license expires — useful for orgs with slow procurement cycles.

```env
LICENSE_EXPIRATION_WARNING_DAYS=30
```

— Exact env var and default given, fully actionable snapshot-20260928

03 **OAuth authorization sessions for agents and workspaces** NEW

70

OAuth authorization sessions add `owner_type` and `owner_id` fields, allowing a managed credential to be authorized for either a user or an agent; sessions now also accept workspace owners so one managed OAuth credential can be shared across a workspace. Agent OAuth connections now show a provider-branded result page after authorization.

— Field names given; no UI path or endpoint specified snapshot-20260928

04 **Thread header dataset and annotation queue actions** NEW

65

Add to dataset and Add to annotation queue buttons were added to the thread header in every thread view (Messages, Turns, and Details), letting a whole thread be added without selecting a run; the add-to-dataset popup for threads now also offers a New Dataset option that creates the dataset empty and adds the selected threads to it.

— Names exact UI views and buttons but no navigation path snapshot-20260928

05 **Bedrock request timeout and retry controls in Playground**

IMPROVED

60

Amazon Bedrock models in the Playground now expose request timeout and max retries options, removing the previous 60-second AWS client default cutoff.

— States before/after behaviour but no exact field names snapshot-20260928

THINNER COVERAGE BELOW

06 **Portable Gateway model configs for Hub-prompt experiments**

NEW

55

Dataset experiments launched from edited Hub prompts save portable Gateway model configurations while preserving variable mappings; saved drafts exclude runtime credentials and remain compatible with standard LangChain SDKs.

— Explains mechanism and compatibility but no config keys shown snapshot-20260928

07 **Sandboxes private ECR IAM role assumption** IMPROVED

50

Commercial private ECR registries in Sandboxes can now assume an AWS IAM role on demand, enabling repository and tag search without expiring saved authorization tokens.

— Names the mechanism but no config or setup steps `snapshot-20260928`

---

08 **MCP connectors use Client ID Metadata Documents** IMPROVED

50

LangSmith MCP connectors use Client ID Metadata Documents when supported, with dynamic client registration retained as a compatibility fallback.

— Names the mechanism but no config steps for readers `snapshot-20260928`

---

09 **IssueBench Opus 5 regression benchmark** NEW

50

IssueBench now includes a benchmark task that compares pre- and post-Opus 5 agent traces for cost and trajectory regressions.

— Names the benchmark and comparison but no usage steps `snapshot-20260928`

---

10 **Claude Fable 5.1 in Playground** NEW

50

Claude Fable 5.1 is now available in the Anthropic, Bedrock, and Vertex AI model selectors in the Playground.

— Names model and selectors, clear starting point `snapshot-20260928`

---

11 **Dedicated gRPC service for trajectory batch reads** IMPROVED

45

Trajectory batch reads now run on a dedicated in-cluster gRPC service, so load from trajectory evaluations no longer competes with UI and API traffic.

— Explains architecture change but nothing a reader can invoke `snapshot-20260928`

---

12 **Thread query server-sent events** NEW

40

Thread query clients can request server-sent events to receive results progressively while preserving cursor pagination.

— Thin description of an API capability, no endpoint given `snapshot-20260928`

---

13 **Workspace switcher accepts workspace IDs** IMPROVED

30

The workspace switcher now accepts workspace IDs in addition to workspace names.

— Bare statement with no further detail `snapshot-20260928`

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read AND download permissions — API requests that previously succeeded with only read permission now receive a forbidden response.

snapshot-20260928 NOTES ↗ LATEST STABLE

LangSmith: Adds `owner_type` and `owner_id` fields to OAuth authorization sessions, allowing a managed credential to be authorized for either a user or an agent.

└──▷ **USE IT**

Route custom model traffic through the LLM Gateway and attribute traces to the correct provider pricing by setting `ls_provider` in the model configuration.

```json
{
  "model": "my-custom-gpt4o",
  "base_url": "https://my-gateway.example.com/v1",
  "ls_provider": "openai"
}
```

Give self-hosted teams earlier notice before a license expires — useful for orgs with slow procurement cycles.

```.env
LICENSE_EXPIRATION_WARNING_DAYS=30
```

Adds `LICENSE_EXPIRATION_WARNING_DAYS` environment variable for self-hosted operators to configure how many days before license expiration the warning banner appears (default: 7).

Adds `ls_provider` field on OpenAI-compatible model configurations in LLM Gateway to populate trace metadata and match custom models to provider-specific pricing.

LLM Gateway fallback chains can now advance to the next model when a candidate does not return response headers within a configured timeout, while active response streams continue after headers arrive.

Commercial private ECR registries in Sandboxes can now assume an AWS IAM role on demand, enabling repository and tag search without expiring saved authorization tokens.

Thread query clients can request server-sent events to receive results progressively while preserving cursor pagination.

+13 more

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read AND download permissions — API requests that previously succeeded with only read permission

now receive a forbidden response.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20260928 NOTES ↗ RANK

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

> Braintrust shipped local secret-redaction plugins and reusable observability templates for the bt CLI, a new Patterns engine that surfaces recurring trace issues, DeepSeek V4.1 Flash as a built-in model, and broader SQL, billing, and log-search improvements, alongside a change to how project vs. organization model providers are resolved.

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

01 Local span-plugin redaction via bt CLI NEW

95

`bt` CLI v0.21.0+ adds `bt trace enable`, `bt trace run`, and `bt trace import` commands for registering JavaScript span plugins that redact secrets, remove sensitive fields, or add tags before traces leave your machine; affected spans are withheld if a plugin fails.

Redact secrets from coding-agent spans locally before they are sent to Braintrust, using a JavaScript plugin registered with the bt CLI.

```
$ bt trace enable my-redact-plugin.js
bt trace run -- node my-agent.js
```

— Names exact CLI commands, version, and runnable example. snapshot-20260928

02 Portable observability templates via bt CLI NEW

90

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

Reuse an existing project's observability setup — facets and Loop automations — by exporting it as a JSON template and applying it to a new project.

```
$ bt observability template export --project my-source-project >
obs-template.json
bt observability template apply --project my-new-project obs-
template.json
```

— Exact commands, version, and runnable export/apply example. snapshot-20260928

03 TypeSafe Jev integration for tracing and judging NEW

80

Trace Jev calls from Python (v0.41.0+) or TypeScript (v3.34.0+) to inspect context, question criteria, and selected answers, and use Jev as the judge in a scorer or classifier with per-choice probabilities and confidence scores (requires data plane v2.15.0+).

— Names languages, versions, and scoring mechanism precisely. snapshot-20260928

04 **Metric columnstore for high-volume Logs pages** IMPROVED 70

Enabling Metric columnstore in project settings stores trace-level metrics in columnar format, speeding up loading and sorting on the Logs page for high-volume projects (requires data plane v2.2.1+).

— Names the settings toggle and performance mechanism. snapshot-20260928

05 **DeepSeek V4.1 Flash built-in model** NEW 70

DeepSeek V4.1 Flash is now a built-in model in playgrounds, prompts, and scorers with multimodal reasoning support, and is accessible via the Braintrust Gateway as `deepseek-v4.1-flash` with no AI provider setup required.

— Names exact gateway model identifier and surfaces it's used in. snapshot-20260928

06 **Per-project usage analytics on billing page** NEW 70

The billing page now shows a usage analytics breakdown by project, with CSV export of logs, scores, and Topics token spend; data is available from September 1, 2026 (requires data plane v2.13.0+).

— Names export format, data scope, and availability date. snapshot-20260928

07 **Streaming Brainstore log search** IMPROVED 70

Brainstore log search now streams results from an asynchronous execution engine as segments finish scanning, covering full-text, phrase, and multi-ANY_SPAN() trace searches including negated clauses (requires data plane v2.13.0+).

— Names search modes and query function but no exact command. snapshot-20260928

08 **Patterns engine for recurring trace issues** NEW 65

Patterns is a scheduled Loop-powered engine that searches project traces for recurring problems and trends, saves findings with supporting traces and suggested fixes, and lets users create a scorer or classifier from any finding (requires data plane v2.13+).

— Describes mechanism and outputs but no exact UI path. snapshot-20260928

09 **Project vs. organization provider precedence change** BREAKING 65

On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models

using their own credentials. Teams relying on project providers to restrict model access or spending should review organization provider model selections and permissions.

— Explains behavior change and gives a concrete migration action. snapshot-20260928

10 **SELECT DISTINCT in Braintrust SQL** NEW 60

Braintrust SQL queries now support `SELECT DISTINCT` on scalar columns, returning one row per unique combination of values (requires data plane v2.15.0+).

— Named syntax and version requirement but no example query. snapshot-20260928

THINNER COVERAGE BELOW

11 **Clipboard copy for span names in trace panels** IMPROVED 45

Clicking a span name in any trace panel now copies it to the clipboard instead of filtering the logs list, available across all trace panel views.

— Clear before/after but no config or command surface. product docs

└──▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials. If you rely on project providers to restrict model access or spending, review organization provider model selections and permissions.

snapshot-20260928 NOTES ↗ LATEST STABLE

Braintrust: Adds `bt trace enable`, `bt trace run`, and `bt trace import` commands to the `bt` CLI (v0.21.0+) for registering JavaScript span plugins that redact secrets, remove sensitive fields, or add tags before traces leave your machine; withholds affected spans if a plugin fails.

└──▷ **TRY IT**

Redact secrets from coding-agent spans locally before they are sent to Braintrust, using a JavaScript plugin registered with the bt CLI.

```
$ bt trace enable my-redact-plugin.js
bt trace run -- node my-agent.js
```

Reuse an existing project's observability setup — facets and Loop automations — by exporting it as a JSON template and applying it to a new project.

```
$ bt observability template export --project my-source-project > obs-template.json
bt observability template apply --project my-new-project obs-template.json
```

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

Adds `SELECT DISTINCT` on scalar columns to Braintrust SQL queries, returning one row per unique combination of values (requires data plane v2.15.0+).

Enables Metric columnstore in project settings to store trace-level metrics in columnar format, speeding up loading and sorting on the Logs page for high-volume projects (requires data plane v2.2.1+).

Adds Patterns: a scheduled Loop-powered engine that searches your project traces for recurring problems and trends, saves findings with supporting traces and suggested fixes, and lets you create a scorer or classifier from any finding (requires data plane v2.13+).

Adds TypeSafe Jev integration: trace Jev calls from Python (v0.41.0+) or TypeScript (v3.34.0+) to inspect context, question criteria, and selected answers; use Jev as the judge in a scorer or classifier with per-choice probabilities and confidence scores (requires data plane v2.15.0+).

+3 more

└─▷ **BREAKING ON UPGRADE**

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials. If you rely on project providers to restrict model access or spending, review organization provider model selections and permissions.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

ai-safe2-framework

Feedback ⓘ 2026-09-27_CLI_0.9.0

NOTES ↗ RANK

Ai-Safe2-Framework provides governance, risk, and compliance controls for securing agentic AI systems and non-human identities.

> AI SAFE² CLI 0.9.0 introduces a full controlled-execution pipeline for its frozen Challenge 001 evaluation, adding pre-registration, bounded execution, independent verification, sealed evidence receipts, and evidence-manifest integration, alongside more explicit failure handling, independent grading, and process security boundaries.

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

01 **Challenge 001 controlled execution workflow** NEW 95

Adds `safe2 challenge plan` to pre-register the evaluator, provider, treatment, execution limits, command-file identities, and optional system identity before any process runs; `safe2 challenge execute` to run the six frozen Challenge 001 scenarios as separate bounded process calls after explicit operator authorization via `--authorize-process-execution`; and `safe2 challenge verify-execution` to recheck the plan, source, normalized run, receipt, command-file digests, and optional identity binding without re-running the evaluator. These commands are backed by the `safe2.challenge-execution-plan.v1` schema, a strict versioned pre-registration contract binding the executable, arguments, provider, treatment, and limits, and the `safe2.challenge-execution-receipt.v1` schema, a sealed record of cross-artifact and per-episode evidence bindings including before-and-after SHA-256 digests for the executable and file arguments. A packaged example is verifiable via `safe2 example verify challenge-controlled-executor`, demonstrating the full planning, execution, verification, reporting, and evidence-manifest workflow.

— Names exact commands, flags and schema identifiers with clear mechanism

2026-09-27_CLI_0.9.0

THINNER COVERAGE BELOW

02 **Evidence manifest support for execution artifacts** IMPROVED 50

Extends `safe2 evidence manifest` to accept execution plans and execution receipts, allowing them to enter the existing AI SAFE² evidence inventory.

— Names the command but gives little mechanism detail 2026-09-27_CLI_0.9.0

03 **Explicit incomplete-episode capture for evaluator failures**

IMPROVED

45

Invalid JSON responses, nonzero exits, timeouts, output byte limits, and incomplete streams from an evaluator process are now captured as explicit incomplete episodes rather than requiring an external reporting convention.

— Describes behavior change but names no config surface 2026-09-27_CLI_0.9.0

04 **Independent state-based grading of evidence** IMPROVED

25

Adds independent state-based grading of recorded evidence, replacing acceptance of the provider's own score.

— Clear before/after but no named surface or mechanism 2026-09-27_CLI_0.9.0

05 **Security boundaries around controlled execution** NEW

25

Adds no-overwrite artifact creation and reduced child-process environment variables as security boundaries around controlled execution.

— Names two mechanisms without further configuration detail 2026-09-27_CLI_0.9.0

2026-09-27_CLI_0.9.0 NOTES ↗ LATEST STABLE

AI SAFE² CLI 0.9.0 adds controlled Challenge 001 execution with pre-registration, independent grading, and sealed evidence receipts.

└──▷ GET THIS VERSION

```
$ git clone --branch 2026-09-27_CLI_0.9.0
https://github.com/CyberStrategyInstitute/ai-safe2-framework.git
# already have the repo? check out this version:
$ git checkout 2026-09-27_CLI_0.9.0
```

Adds `safe2 challenge plan` command to pre-register the evaluator, provider, treatment, execution limits, command-file identities, and optional system identity before running any process.

Adds `safe2 challenge execute` command to run the six frozen Challenge 001 scenarios as separate bounded process calls after explicit operator authorization via `--authorize-process-execution`.

Adds `safe2 challenge verify-execution` command to recheck the plan, source, normalized run, receipt, command-file digests, and optional identity binding without re-running the evaluator.

Introduces the `safe2.challenge-execution-plan.v1` schema as a strict, versioned machine-readable pre-registration contract binding the executable,

arguments, provider, treatment, and limits.

Introduces the `safe2.challenge-execution-receipt.v1` schema as a sealed record of cross-artifact and per-episode evidence bindings, including before-and-after SHA-256 digests for the executable and file arguments.

+5 more

WAS THIS USEFUL? 👍 👎

Mend.io Mend AI

Feedback ⓘ (27-September-2026) NOTES ↗

RANK

Mend AI is an AI security service that discovers AI components and helps teams govern their use in software applications.

> This window's headline is Mend AI extending Suppress and Review triage down to the individual finding level, alongside CLI, SAST and Renovate updates including a new console-streaming flag, Bun package manager support, macOS metadata exclusions, and expanded Renovate versioning scheme handling.

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

01 **Console streaming for CLI SAST scans** NEW 75

`mend code` SAST scans now support the `--print-to-console` flag, streaming CLI logs directly to stdout as the scan runs in addition to the log file — a capability previously only available for `mend dep`.

Tail SAST scan progress in real time during CI runs without having to read a separate log file.

```
$ mend code --print-to-console
```

Stream SAST scan output live to the terminal during CI runs so you can watch findings appear without waiting for the log file.

```
$ mend code --print-to-console
```

— Names exact flag and prior scope; gives runnable command. product docs

02 **Versioning scheme support in Renovate** IMPROVED 75

Mend Renovate adds `gomod` versioning to support Go module version resolution, with pseudo-version awareness added to the `gomod` versioning handler; extends the downgrade guard to apply across every versioning scheme rather than only select ones; and accepts PEP 440 specifiers in Poetry versioning instead of relying on a lookup special case.

— Names all four constituent versioning changes with specifics. product docs

03 **Bun package manager support** NEW 65

Mend CLI now supports the Bun package manager for JavaScript projects, with automatic detection of direct and transitive dependencies in projects using `package.json` alongside `bun.lock` or `bun.lockb` files.

— Names files detected but no command shown. product docs

04 **macOS metadata exclusion in SAST scans** IMPROVED 65

Default exclusions in `mend code` CLI scans now automatically ignore `/_MACOSX/..*` metadata files across all SAST scan engines, preventing macOS system files from interfering with the TreeSitter parser on Windows environments — applied even when default exclusions are otherwise disabled.

— Explains mechanism and scope but is automatic, no user action. product docs

05 **CWE-117 log forging remediation update** IMPROVED 65

AI-generated remediation suggestions for CWE-117 (Log Forging) in Python now neutralize CR and LF characters instead of applying HTML escaping, sanitizing all client-controlled values within log statements.

— Clear before/after mechanism but no user-facing action. product docs

THINNER COVERAGE BELOW

06 **Finding-level Suppress and Review triage** IMPROVED 55

Suppress and Review triage actions now extend to the individual finding level, enabling security teams to triage specific findings within an entity independently rather than only at the entity level.

— Describes scope change but no exact UI path or endpoint. (27-September-2026)

> (27-September-2026) NOTES ↗ LATEST STABLE
>
> Mend AI extends Suppress and Review triage actions to individual finding level for more precise control.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

DeepSeek V3

Feedback ⓘ price cut · Sep 27, 2026 MODEL ↗

DeepSeek V3 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-chat): $0.32 → $0.2574 in, $0.89 → $1.0287 out per million tokens (-20%), first seen 27 Sep 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4 Pro 0813

Feedback ⓘ price cut · Sep 27, 2026

MODEL ↗ STABLE

DeepSeek V4 Pro 0813 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4-pro): $0.6116 → $0.348 in, $1.2232 → $0.696 out per million tokens (-43%), first seen 27 Sep 2026.

▾ 1 more release in this issue · 2026-09-27

> price increase · Sep 27, 2026 MODEL ↗ STABLE
>
> Price increase on OpenRouter (deepseek/deepseek-v4-pro-0813): $0.3458 → $0.245 in, $1.0375 → $3.50 out per million tokens (+237%), first seen 27 Sep 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4.1 Flash

Feedback ⓘ price cut · Sep 27, 2026

MODEL ↗

DeepSeek V4.1 Flash is a multimodal model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4.1-flash): $0.099 → $0.035 in, $0.60 → $0.29 out per million tokens (-65%), first seen 27 Sep 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M2

Feedback ⓘ price cut · Sep 27, 2026 MODEL ↗

MiniMax M2 is a reasoning model from MiniMax.

Price cut on OpenRouter (minimax/minimax-m2): $0.30 → $0.255 in, $1.20 → $1.02 out per million tokens (-15%), first seen 27 Sep 2026.

WAS THIS USEFUL? 👍 👎

MiniMax M2.7

Feedback ⓘ price cut · Sep 27, 2026 MODEL ↗

MiniMax M2.7 is a reasoning model from MiniMax.

Price cut on OpenRouter (minimax/minimax-m2.7): $0.30 → $0.21 in, $1.20 → $0.84 out per million tokens (-30%), first seen 27 Sep 2026.

WAS THIS USEFUL? 👍 👎

◆ Frontier Models

xAI Grok API

Feedback ⓘ snapshot-20260928 NOTES ↗ RANK

Official SpaceXAI API documentation. Build with Grok: chat, image and video generation, voice, tool calling, and structured outputs.

> Grok API added a `safety_identifier` field to help attribute policy violations to individual end users rather than API keys, spanning multiple request surfaces.

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

01 **Per-user safety identifier field** NEW 71

Adds a `safety_identifier` request field to Chat Completions, the Responses API, deferred chat completions, the Batch API, and `gRPC GetCompletionsRequest`, allowing policy violations to be attributed to a specific end user rather than to your API key.

— Names field and five API surfaces, but no usage example given. snapshot-20260928

> snapshot-20260928 NOTES ↗ LATEST STABLE
>
> Grok API adds `safety_identifier` field for per-user policy attribution across Chat Completions, Responses, Batch, and gRPC APIs.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 28, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue