THE AI TOOLCHAIN

NO. 021

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 9, 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 266 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 36 tools

▾ // THE BRIEF HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: Opus 5 | GPT-5.5

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. vLLM adds request-queue admission control and per-request speculative-decoding metrics, so overload gets

vim ? help

shed at the door instead of degrading everyone. JAX lets you write CUDA kernels in Python that still `jax.jit`, autodiff and shard. Fireworks adds per-user spending limits and bring-your-own provider keys, capping spend before the invoice.

DEPLOY

Shed load at the gate instead of letting a queue melt tail latency

vLLM v0.29.0 gates entry to the request queue rather than accepting everything and letting time-to-first-token slide for every in-flight request, and its per-request speculative-decoding metrics finally show whether draft tokens are being accepted or wasted on your traffic mix. mlx-serve adds disk-backed prefix caching so a long shared system prompt survives process restarts instead of being recomputed.

vLLM · mlx-serve

BUILD

Write a custom CUDA kernel in Python and keep the rest of your JAX program intact

JAX's CuTe DSL integration means a hand-written kernel composes with `jax.jit`, autodiff, sharding and export instead of forcing a foreign-function escape hatch that breaks tracing. Accelerate v1.15.0 adds CPU offload for FSDP2 activation checkpointing and per-rank sharded checkpointing, buying back memory on the runs that were just barely not fitting.

JAX · Accelerate

GOVERN

Cap what a single user or key can spend before finance finds out

Fireworks adds per-user spending limits, bring-your-own provider keys and SSO/SCIM provisioning, so a runaway agent loop hits a ceiling rather than an invoice, and identity comes from your IdP instead of a shared console login. Mem0's DeepSeek plugin now puts cross-user memory reads behind an explicit operator flag, closing a path where one tenant's recalled context reaches another.

Fireworks AI · Mem0

Does **Opus 5** read better? 👍 👎

BUILD

◆ AI Coding Agents

StackBlitz bolt.new

Feedback ⓘ snapshot-20260909 NOTES ↗

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

Bolt.new adds a visual editor in the preview pane — edit text, colors, and styles before committing any tokens.

└──▷ HOW TO FIND IT

Iterate on UI appearance without spending tokens — use the Select tool to tweak colors and text, review all changes in the batch, then commit once when satisfied.

📍 In the chatbox, click the Select tool, then click any element in the preview to edit its text, color, font size, formatting, or alignment. Review the collected edits above the chatbox, then click `Save changes` to apply them to your project code.

Adds a Select tool in the chatbox for making visual edits — text, colors, font size, formatting, and alignment — directly in the preview, with changes batched and applied to project code only when `Save changes` is clicked.

WAS THIS USEFUL? 👍 👎

Cline

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

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

> Cline's desktop app adds authenticated remote Hub SDK connectivity, default web search, imported-session summarization from other coding agents, and a string of reliability and performance fixes to checkpoints, patching, and session history.

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

01 Imported session summaries from other coding agents NEW

62

Sessions imported from Claude Code, Codex, and opencode now display their source in chat and summarize the foreign history on the first resumed turn, showing a 'Summarizing the imported <tool> history...' indicator while running.

– Names sources and exact indicator text with mechanism desktop-v0.0.24

THINNER COVERAGE BELOW

02 apply_patch Add File no longer overwrites silently IMPROVED

58

`apply_patch` Add File now refuses to overwrite an existing file instead of silently replacing it.

– Names exact tool and action, clear before/after behavior desktop-v0.0.24

03 Checkpoint hashing performance overhaul IMPROVED

55

Checkpoints no longer re-hash every untracked file before each message; a single snapshot index is kept per session, so from the second turn cost drops to roughly git process overhead.

– Clear before/after mechanism and performance claim, no user action needed

desktop-v0.0.24

04 Web search on by default outside YOLO mode IMPROVED

50

Web search is now enabled by default outside YOLO mode, and tool settings fail closed if they cannot be loaded.

– Explains behavior change but no config key to toggle it desktop-v0.0.24

05 Custom title bar on Windows NEW

50

Adds a custom title bar on Windows with caption controls that follow the compact title-bar height in narrow windows and stay above overlays.

— Describes UI behavior across window sizes, no config surface `desktop-v0.0.24`

06 **@ mention no longer indexes home or filesystem root** IMPROVED

43

Typing an `@` mention no longer indexes the home directory or filesystem root.

— Names exact trigger and fixed scope, minimal further detail `desktop-v0.0.24`

07 **Session-history token counts and costs backfilled** IMPROVED

42

Token counts and costs now fill in for every visible session-history page, hydrated on demand with reads re-run when session status changes.

— Explains mechanism but no UI path or command given `desktop-v0.0.24`

08 **SDK support for authenticated remote Hubs** NEW

33

Adds SDK support for connecting to authenticated remote Hubs.

— Names the capability but no connection mechanism or API detail `desktop-v0.0.24`

WAS THIS USEFUL?  

OpenAI Codex CLI

Feedback ⓘ 3 RELEASES · 2026-09-08 → 2026-09-09

NOTES ↗ CODE ↗ RANK

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

> Codex CLI's alpha releases add live WebRTC voice conversations and controls to the TUI, a configurable memory v2 storage system, an explicit `app-server daemon update` command, and raise Guardian's action review limit to 200,000 bytes, alongside AWS Bedrock credential export, MCP reconnect/verification signals, and assorted TUI and telemetry improvements.

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

01 **`app-server daemon update` subcommand** NEW 90

Adds `app-server daemon update` subcommand to update the standalone Codex installation and restart the managed daemon, letting users trigger an immediate update without waiting for the automatic update cycle.

Trigger an immediate update of the standalone Codex installation and restart the managed daemon without waiting for the automatic update cycle.

```
$ codex app-server daemon update
```

— Exact runnable subcommand with behaviour explained rust-v0.154.0-alpha.7

02 **Guardian action review byte limit raised** IMPROVED 80

Raises `GUARDIAN_MAX_ACTION_BYTES` from 8,000 to 200,000 bytes, unlocking Guardian review of much larger agent actions and tool outputs; exec stdin approvals remain separately capped at 8,000 bytes via the distinct `MAX_STDIN_APPROVAL_BYTES` constant.

— Names both constants and exact byte values rust-v0.154.0-alpha.10.2 rust-v0.154.0-alpha.6.1

03 **Live voice conversations and controls in TUI** NEW 70

Adds live WebRTC voice conversations to the TUI with a dedicated composer strip for voice controls, a configurable voice mute shortcut, recording activity indicators, and split-flap tile animations for live voice transcripts.

— Multiple named UI elements but no exact key path rust-v0.154.0-alpha.7

04 **Configurable memory versions (memory v2)** NEW 70

Adds configurable memory versions with isolated storage (memory v2), including dual writing, dedicated consolidation and read prompts, and summary-only extraction.

– Explains mechanism and storage behaviour in detail `rust-v0.154.0-alpha.7`

05 **App-server daemon configuration and status visibility**

IMPROVED

65

Shows configured app-server updater settings (`auto_update_enabled`, `update_interval_minutes`) in `codex doctor` output; makes app-server daemon automatic updates configurable and makes the managed app-server shutdown grace period configurable; also warns when the connected Codex service is older than the CLI, with older app-server version notices now configurable in the TUI.

– Names doctor keys but not the new config keys themselves `rust-v0.154.0-alpha.7` `rust-v0.154.0-alpha.10.2`

06 **Voice session and TUI start telemetry metrics** NEW

65

Records voice session lifecycle metrics under `codex.voice.session.*` (starts, connections, failures, endings, active duration), and tags `codex.tui.start` metrics with `terminal_name` and `multiplexer` attributes (`tmux`, `zellij`, or `none`).

– Names metric prefixes and tag values precisely `rust-v0.154.0-alpha.10.2`

07 **Worktree delete action in managed browser** NEW

60

Adds a worktree delete action with a confirmation dialog to the managed worktree browser, including owner thread titles, relative update times, and archived/unavailable status.

– Describes UI mechanism and fields shown `rust-v0.154.0-alpha.10.2`

THINNER COVERAGE BELOW

08 **MCP OAuth reconnect signal in TUI** NEW

45

Surfaces MCP reconnect signals in the TUI when expired OAuth tokens cannot be refreshed.

– Describes trigger but no exact UI path `rust-v0.154.0-alpha.10.2`

09 **AWS Bedrock credential export commands** NEW

40

Adds AWS credential export commands for Amazon Bedrock integration.

– No command name or flag given `rust-v0.154.0-alpha.10.2`

10 **macOS Secure Enclave user verification** NEW 40

Adds macOS user verification with Secure Enclave signing.

— Named mechanism but no usage steps `rust-v0.154.0-alpha.7`

11 **MCP user verification for host-owned apps server** NEW 40

Adds MCP user verification in the TUI, restricted to the host-owned apps MCP server.

— Scope named but no config or command `rust-v0.154.0-alpha.7`

12 **Configurable credential providers for network proxy** NEW 30

Adds configurable credential providers to the network proxy.

— No provider names or config keys given `rust-v0.154.0-alpha.10.2`

13 **TUI turn timestamps and action grouping** IMPROVED 30

Shows completion timestamps after successful TUI turns and groups adjacent computer actions in the TUI display.

— Cosmetic UI change with minimal detail `rust-v0.154.0-alpha.7`

14 **Image attachments in agents overview** NEW 25

Supports image attachments in agents overview background tasks.

— Bare capability statement, no mechanism `rust-v0.154.0-alpha.10.2`

15 **Signed voice resources on macOS** IMPROVED 20

Bundles signed voice resources in macOS releases.

— One-line packaging note, no further detail `rust-v0.154.0-alpha.10.2`

WAS THIS USEFUL?  

SST OpenCode

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

RANK

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

> OpenCode v1.18.30 introduces a new system prompt for GPT-6 models and adds configurable reasoning effort for GitLab-hosted GPT and Claude models.

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

01 **Reasoning effort variants for GitLab-hosted models** NEW 60

Adds reasoning effort variants for GitLab-hosted GPT models via `reasoningEffort` and for Claude models via adaptive `thinking`.

— Names exact config surfaces but no usage detail or defaults v1.18.30

THINNER COVERAGE BELOW

02 **Astra system prompt for GPT-6 models** NEW 23

Adds the Astra system prompt for GPT-6 models.

— Only named in summary, no mechanism or usage given v1.18.30

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ 1 RELEASE · 2026-09-08

NOTES ↗ CODE ↗ RANK

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

> Claude Code v2.1.265 adds directory-based plugin loading with hot-reload, a 1 GB cap on saved tool results, faster worktree startup, and improved workflow and slash-command UIs, plus VS Code session archiving.

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

01 **Plugin directory loading with hot-reload** NEW 85

Pointing `--plugin-dir` at a folder of plugins loads each child folder containing a manifest automatically, and plugins added or removed while Claude Code is running are picked up live without a restart.

Point Claude Code at a plugins directory so all sub-plugins load at once and any you drop in or remove are picked up without restarting.

```
$ claude --plugin-dir ./my-plugins/
```

— Named flag with clear mechanism and runnable example v2.1.265

02 **Richer /workflows agent detail view** IMPROVED 60

In `/workflows`, tool calls are now marked running, failed, or done, the subagent's task list is shown when available, and pressing Enter unfolds listed calls to show their inputs and results.

— Names command and exact interaction (Enter key) v2.1.265

THINNER COVERAGE BELOW

03 **Expanded telemetry fields for gateway sessions** IMPROVED 55

Claude Desktop and Cowork now send `user.email` and `user.groups` in the telemetry pushed through a Claude apps gateway, matching what terminal sessions already report.

— Names exact fields but no user action possible v2.1.265

04 **Faster worktree startup on large repos** IMPROVED 55

`--worktree` startup is improved on large repositories: the new worktree is now checked out in parallel, requiring git 2.32 or later.

— Names flag, mechanism and version requirement v2.1.265

05 **Improved slash command and skill matching** IMPROVED

55

Slash commands typed mid-prompt now show matches in a list, with Tab opening the list outside fullscreen instead of only showing a single suggestion, and a plugin skill is now findable by its bare name.

— Names Tab behavior but limited mechanism detail v2.1.265

06 **Auto-archive inactive sessions in VS Code** NEW

55

A new Archive inactive sessions setting in the VS Code extension automatically archives sessions inactive for a set period, defaulting to 14 days.

— Names the setting and default but no path given v2.1.265

07 **1 GB cap on saved tool results** IMPROVED

45

Tool results saved to disk are now capped at 1 GB, and the in-conversation preview indicates when a saved file was truncated due to hitting this cap.

— Concrete limit named but no configurable surface v2.1.265

WAS THIS USEFUL?  

Diagram Design

Feedback ⓘ changes since 2026-08-12 CODE ↗

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

Diagram Design adds ten editorial diagram types, native Droid plugin packaging, named client profiles, accessible-SVG contract, and animated semantic-pattern examples.

└──▷ **TRY IT**

Catch SVGs that will fail screen-reader accessibility — for example before merging a PR that adds a new diagram — by running the a11y lint gate locally.

```
$ python scripts/lint-skin.py --all --baseline
```

Adds `role="img"`, a resolving `aria-labelledby`, and first-child `<title>` / `<desc>` slots to `template.html`, `template-dark.html`, `template-full.html`, and `template-terminal.html` as an enforced accessible-SVG contract.

Adds an `a11y` category to `scripts/lint-skin.py` that rejects a diagram SVG without a resolvable accessible name.

Adds `scripts/verify-beeswarm.py` to verify that every beeswarm dot sits at its declared value on the shared linear scale and that no two dots overlap.

Adds `scripts/test-verify-sankey.py` with adversarial fixtures covering ribbon narrowing, node volume loss, stage-flow shrinkage, label/bar mismatches, and missing shipped variants.

Ships ten editorial diagram types in one release (`feat(types): ship ten editorial diagram types`).

+5 more

WAS THIS USEFUL? 👍 👎

Block Goose

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ CODE ↗

RANK

Goose is an open-source AI agent with a desktop app and CLI for coding and other automated tasks.

> Goose v1.50.0 expands model provider support with GPT-6 Astra and AIML API, splits out ACP npm packages, and adds Kotlin GDK configuration for Databricks AI Gateway.

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

01 **Separate npm packages for ACP client and binary** IMPROVED

50

The goose ACP client and the `acp` binary are now published as separate npm packages.

— Names exact packages, no install instructions v1.50.0

02 **New provider support: GPT-6 Astra and AIML API** NEW

35

Goose adds GPT-6 Astra models as a selectable provider and integrates AIML API as a new provider option.

— Names two providers but no config details v1.50.0

03 **Kotlin GDK configuration for Databricks AI Gateway path** NEW

30

Enables Kotlin callers in the GDK to configure the Databricks AI Gateway path.

— States capability but no config key or example v1.50.0

04 **Upgraded MCP version preference** IMPROVED

10

Upgrades MCP version preference to the latest available.

— Bare statement with no version number or mechanism v1.50.0

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

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

> Letta Code shipped a single small update in this window: the ability to cancel Monitors from connected clients via the listener.

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

01 **Cancel Monitors from connected clients** NEW 30

Connected clients can now cancel Monitors through the listener.

— No mechanism, command, or config detail given, just a bare capability. v0.31.14

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ

5 RELEASES · 2026-09-08 → 2026-09-09 NOTES ↗ RANK

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

> Qwen Code's biggest addition is a one-command `qwen serve` remote session with QR pairing, alongside a maturing SDK — multi-modal input, session export, experimental LSP support, a Java SDK alpha, and settings-based environment configuration — and Agent Skills promoted to stable. The window also brought sweeping web-shell, channel, and daemon/session management additions plus several breaking changes: the `disable*` → `enable*` settings rename, retirement of `@qwen-code/webui`, and removal of the Smart Edit tool and `read_many_files`.

└──▷ **WHAT SHIPPED · 44 FEATURES** most completely described first ? what's the number?

01 **One-command remote serve with QR pairing** NEW 80

`qwen serve` starts a remote session with a single command, generating an access token, enabling same-origin shell access, and printing a pairing QR code so a client can connect easily.

— Names the exact command and its mechanism, no full flag reference v0.23.2

02 **Environment variables and extra_body in SDK settings** NEW 75

Adds a `settings.env` field for configuring environment variables directly in the settings file instead of the shell, and adds `extra_body` support for OpenAI-compatible providers to pass through provider-specific fields not covered by the standard API, such as `enable_thinking` or extended thinking budgets.

Set provider-specific environment variables in settings rather than in the shell, so every Qwen Code session picks them up automatically.

json

```
{
  "settings": {
    "env": {
      "OPENAI_API_KEY": "sk-...",
      "OPENAI_BASE_URL": "https://api.example.com/v1"
    }
  }
}
```

Pass provider-specific parameters not covered by the standard API surface when using an OpenAI-compatible endpoint.

```json
{
  "settings": {
    "extra_body": {
      "enable_thinking": true
    }
  }
}
```

Set provider API keys or proxy variables for a session without exporting them to the shell environment.

```json
{
  "env": {
    "OPENAI_API_KEY": "sk-...",
    "HTTPS_PROXY": "http://proxy.internal:3128"
  }
}
```

Set provider API keys and custom base URLs directly in settings so every team member picks them up without shell exports.

```json
{
  "settings": {
    "env": {
      "OPENAI_API_KEY": "sk-...",
      "OPENAI_BASE_URL": "https://your-proxy.example.com/v1"
    }
  }
}
```

Pass provider-specific parameters (e.g. extended thinking budget) through to an OpenAI-compatible endpoint that supports them.

```json
{
  "extra_body": {
    "thinking": {
      "type": "enabled",
      "budget_tokens": 8000
    }
  }
}
```

— Exact config keys with runnable JSON examples across three releases

sdk-typescript-v0.1.11 sdk-typescript-v0.1.10 sdk-typescript-v0.1.9

03 **Managed auto-memory setting** IMPROVED 75

Managed memory availability now respects the `memory.enableManagedAutoMemory` setting, so hosts that disable managed auto memory no longer see remember/dream requests admitted or managed-memory instructions in the system prompt.

— Exact key and before/after behavior stated sdk-typescript-v0.1.10 sdk-typescript-v0.1.9

04 **Daemon session fencing and controller tokens** IMPROVED 75

Adds session turn navigation protocol support in the daemon and SDK for bounded turn indexing, snapshots, and transcript paging; adds a session resource catalog to the serve layer; allows concurrent standalone daemons with session fencing; fences Conversations writers with mandatory session leases; adds a user-minted controller token to drive a session without per-message review; and reaps owned worktrees automatically when daemon sessions are deleted.

— Detailed mechanism across several daemon features, no exact API v0.23.1

05 **Web-shell UI and workflow visibility upgrades** IMPROVED 70

Adds opt-in browser task notifications in the web shell and separates Plan workflow permissions from execution permissions there; adds continuous history and compact turn navigation to the web-shell transcript, a context usage tab to the right sidebar, headless global turn navigation, the ability to bridge a browser-granted local directory into a session, visualization and management of dynamic workflow runs, live status for running subagents in the transcript, and customizable artifact icons.

— Names many discrete UI additions but no exact navigation paths v0.23.1 v0.23.2

06 **Session ID propagation to providers** NEW 70

Expands `${session_id}` in per-provider `customHeaders` configuration and sends the session ID to Routify endpoints, letting downstream providers correlate requests with a specific session.

Forward the active session ID to a Routify-compatible provider by embedding `${session_id}` in a custom request header.

```yaml
providers:
  my-provider:
    baseUrl: https://proxy.example.com
    customHeaders:
      X-Session-ID: "${session_id}"
```

— Named config field with a runnable YAML example v0.23.1

07 **Clickable OSC 8 links in VP mode** IMPROVED 70

Enables opening OSC 8 links with a plain click in VP mode and adds a right-click menu with Open Link, Copy Link Address, and Copy Selection.

— Exact UI behavior and menu items named v0.23.1

08 **Smart Edit tool and read_many_files removed** BREAKING 65

Removes the Smart Edit tool and `ClearcutLogger`, and removes the `read_many_files` tool, replacing it with a `readManyFiles` utility available only as a user `@ -command`.

— Names the removed and replacement surfaces clearly sdk-typescript-v0.1.11 sdk-typescript-v0.1.10

09 **Custom output styles directory** NEW 65

Loads custom output styles from `~/.qwen/output-styles` and `.qwen/output-styles` directories.

Apply a custom output style to all sessions by dropping a style file into the user-level output-styles directory.

```
$ mkdir -p ~/.qwen/output-styles
cp my-team-style.json ~/.qwen/output-styles/
```

— Exact paths with a runnable install example v0.23.1

10 **Agent Skills system reaches stability** IMPROVED 60

Promotes Agent Skills from experimental to stable, adds symlink support to the skill manager, adds a built-in `zvec-grep` installer to the skills system, and adds a workspace-scoped Skills runtime to the serve layer.

— Several named additions to one module, no usage steps v0.23.1 sdk-typescript-v0.1.10

THINNER COVERAGE BELOW

11 **DingTalk channel permission cards and controls** IMPROVED 55

Tool permission requests are now presented in DingTalk via native interactive cards, background agent aggregation is now optional in the DingTalk channel, and the channel shows dynamic lifecycle tags.

— Describes what changed but not how to configure it v0.23.1 v0.23.2

12 **Workflow run tracking and continuation prompts** IMPROVED 55

Journals failed agents and settles agent failures to null in workflow runs, carries budget figures and progress guidance in continuation prompts for goal tracking, and returns run handles and persists inline scripts from workflows.

— Explains mechanism but no concrete API or field names v0.23.1 v0.23.2

13 **ACP subagent tracking fields** IMPROVED 55

Adds `parentToolCallId` and `subagentType` fields for ACP subagent tracking, and makes subagent execution visible via parent `tool_call` updates in ACP.

— Named fields, but no example payload shown v0.23.1 sdk-typescript-v0.1.10

14 **Settings renamed from disable* to enable*** BREAKING 55

The `disable*` settings are renamed to `enable*` (negative naming replaced with positive naming); any configuration that used `disable*` keys will stop working.

— Clear migration note but no full key mapping listed sdk-typescript-v0.1.11

15 **Inline tool-call argument display** NEW 55

Adds `ui.showToolCallArgs` config setting to render tool-call arguments inline in the UI.

Render tool-call arguments inline in the transcript so you can audit exactly what the agent is invoking during a session.

```yaml
ui:
  showToolCallArgs: true
```

— Exact config key with an example toggling it v0.23.1

16 **Channel worktree tasks, resets, and filtering** NEW 55

Adds worktree-isolated named tasks to channels, implements named-session worktree reset for channels, adds a shared multiline instructions field to channel management, and adds message prefix filtering to channels.

— Names four additions but no usage syntax v0.23.1

17 **Goal safeguards: auto-pause and token spend** NEW 55

Pauses a Goal automatically after three autonomous turns that make no progress, lets operators configure the checkpoint verifier timeout for a full claim list, and shows Goal token spend against its allowed window in the web-shell.

— Concrete thresholds named but no config keys given v0.23.1

18 **Model reasoning effort configuration** NEW 50

Supports GPT-5 and GPT-6 reasoning effort configuration, and adds a core configuration option to configure model reasoning capabilities more broadly.

— Names the models and capability but not the exact config key v0.23.1 v0.23.2

19 **`contextWindowSize` configuration key** NEW 50

Adds a `contextWindowSize` configuration key to control the model context window size directly from settings.

— Names the exact key but no usage example given sdk-typescript-v0.1.10 sdk-typescript-v0.1.9

20 **Session export to Markdown and HTML** NEW 50

Adds an `export` command for session history supporting markdown and HTML output formats.

— Named command and formats, no invocation syntax shown sdk-typescript-v0.1.10 sdk-typescript-v0.1.9

21 **Coding Plan configuration and auth mode** NEW 50

Adds Coding Plan configuration management and a Coding Plan authentication mode with a unified AuthDialog.

– Names the feature and UI element but no config keys sdk-typescript-v0.1.11 sdk-typescript-v0.1.10

22 **Size-triggered microcompaction for long sessions** IMPROVED 50

Size-triggered microcompaction now clears old tool results toward a low watermark, preserving provider prompt-cache reuse across long tool-heavy sessions.

– Mechanism explained but no configuration knob given sdk-typescript-v0.1.10

23 **Mem0 auto-recall and daemon memory writes** NEW 50

Adds opt-in auto recall for administrator-owned Mem0 dialects in external-context, and adds daemon memory writes and daemon memory deletion to external-context.

– Names the memory operations but not how to enable them v0.23.1

24 **Prompt-cache control for Anthropic generator** IMPROVED 50

Adds `cache_control` for system and last user text message in the Anthropic content generator to improve prompt-cache reuse.

– Names the field and target messages, no example call sdk-typescript-v0.1.9

25 **TPM throttling and retry handling** IMPROVED 45

Adds TPM throttling error handling with a 1-minute retry delay, and improves retry logic for 429 and 5xx errors.

– Names error codes and delay but no config surface sdk-typescript-v0.1.11 sdk-typescript-v0.1.10

26 **Default web search for ModelStudio users** IMPROVED 40

Enables the built-in `web_search` tool by default for ModelStudio Standard and Token Plan users.

– Names the tool and audience but no toggle documented v0.23.2

27 **Multi-modal input across content generators** NEW 40

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

– States scope but no mechanism or example sdk-typescript-v0.1.10

28 **Debug output routed to logfile-first logger** IMPROVED 40

Refactors debug mode to route console output to a logfile-first `debugLogger`.

– Names the new logger but no usage instructions sdk-typescript-v0.1.11

29 **Model selection through ACP in VS Code companion** NEW

40

Adds support for model selection through ACP in the VS Code IDE companion.

— Names the surface but no steps to use it `sdk-typescript-v0.1.11`

30 **IPC peer message overflow protection** IMPROVED

35

Drops IPC peer messages that arrive faster than a session can process them and notifies the sender once, preventing session overload.

— Clear behavior, no configuration surface named `v0.23.2`

31 **Zed extension for agent server** NEW

35

Adds a Zed extension for the Qwen Code agent server.

— Names the integration but no install steps `sdk-typescript-v0.1.10`

32 **`todo_write` made opt-in** IMPROVED

35

Makes `todo_write` opt-in via core configuration.

— Names the tool but not the config key needed to enable it `v0.23.1`

33 **Review system prose and recall audits** NEW

35

Adds prose-execution and counter-frame audits to the review system, and adds a low-effort recall floor to the review system.

— Thin description with no mechanism or config `v0.23.1`

34 **OpenTUI parity for dialogs and composer** IMPROVED

35

Adds OpenTUI parity closeout covering dialogs, composer, and shell mode to the CLI.

— Names covered areas but no specifics of the parity work `v0.23.1`

35 **`@qwen-code/webui` package retired** BREAKING

35

The `@qwen-code/webui` package has been retired and is no longer available.

— Names the package but gives no migration path `v0.23.1`

36 **Java SDK (alpha)** NEW

35

Adds Java SDK (alpha) for programmatic access to Qwen Code.

— Names the SDK and stage but no usage example `sdk-typescript-v0.1.9`

37 **MCP tool progress updates in TUI/SDK** NEW

30

Adds MCP tool progress update support in TUI and SDK mode.

— Names surfaces but no mechanism explained `sdk-typescript-v0.1.10`

38 **Large-paste placeholder and macOS enter-submit fix** IMPROVED

30

Adds large-paste placeholder and fixes enter-submit on macOS for improved TUI paste handling.

— Describes the fix but no reproduction detail `sdk-typescript-v0.1.9`

39 **Experimental LSP support** NEW

25

Adds experimental LSP (Language Server Protocol) support for code intelligence.

— Bare mention with no configuration detail `sdk-typescript-v0.1.10`

40 **Concurrent runner for batch CLI execution** NEW

25

Adds a concurrent runner for batch CLI execution.

— One-line mention with no usage detail `sdk-typescript-v0.1.10`

41 **UTF-8 BOM preservation on file edits** IMPROVED

25

Preserves UTF-8 BOM when editing files.

— Single-line fix with no further detail `sdk-typescript-v0.1.11`

42 **Source information tracking in telemetry logs** NEW

25

Adds source information tracking in telemetry logs.

— Bare description, no fields or format given `sdk-typescript-v0.1.11`

43 **Configurable release download base URL** NEW

25

Adds support for a configurable release download base URL.

— Bare mention, no config key named `v0.23.1`

44 **Hint for installing external source extensions** NEW

15

Adds a hint for installing external source extensions.

— Bare one-line mention with no detail `sdk-typescript-v0.1.11`

└──▷ **BREAKING ON UPGRADE**

! The `disable*` settings are renamed to `enable*` (negative naming replaced with positive naming) — any config that used `disable*` keys will stop working.

! The `read_many_files` tool is removed; functionality moves to a `readManyFiles` utility available only as a user `-command`.

! The `@qwen-code/webui` package has been retired and is no longer available.

WAS THIS USEFUL?  

DeepSeek Harness

Feedback ⓘ 2 RELEASES · 2026-09-08 → 2026-09-09

CODE ↗ RANK

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

> DeepSeek Harness reworked its client UI around a new Sidebar with panels, download cards and a redone stats view, added weighted pull request approvals, bounded byte-range file reads across `fs-local` and `fs-e2b`, a dual-face file resource addressing API, and CLI profile templates, alongside file-editor plugin mounts and a unified feedback dialog.

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

01 File editor plugin mounts in sdk-minimal profile NEW

90

Adds `@deepseek-ai/dsh-fs-local` and `@deepseek-ai/dsh-tool-str-replace-editor` plugin mounts to the `sdk-minimal` profile's patch layer, giving agents an explicitly mounted file editor for continuation benchmarks. Profiles can insert these via an `insert` block naming `id: fs-local` / `name: '@deepseek-ai/dsh-fs-local'` and `id: str-replace-editor` / `name: '@deepseek-ai/dsh-tool-str-replace-editor'`.

Mount the local filesystem and str-replace editor plugins in your own `sdk-minimal` profile patch layer to give the agent file-editing capability.

yaml

```yaml
- insert:
    - id: fs-local
      name: '@deepseek-ai/dsh-fs-local'
    - id: str-replace-editor
      name: '@deepseek-ai/dsh-tool-str-replace-editor'
```

— Names exact plugin ids and includes a runnable config snippet dsh-v0.1.5-alpha.2

02 Sidebar overhaul with panels and file tree in client UI

IMPROVED

70

Replaces the former Details panel with a Sidebar featuring tab navigation, lazy workspace file tree tabs, paged file tabs with retained reader state, and a fullscreen shell. Also adds global sidebar panels supporting extensible panel entries and panel navigation, plus extensible document previews and sidebar defaults.

— Names UI components but gives no exact navigation path dsh-v0.1.5-alpha.2 dsh-v0.1.5-alpha.1

03 **Bounded byte-range file reads in fs-local and fs-e2b** NEW

70

Adds bounded byte-range reads across local and E2B sandbox filesystems (`fs-local` and `fs-e2b`), enabling partial file access without loading full content.

— Names both filesystem plugins and the behaviour change `dsh-v0.1.5-alpha.1`

04 **Dual-face file resource addressing API** NEW

65

Adds a dual-face file API with Host-resolved resources and a Client resource registry with retained subscriptions for workspace file operations, plus Session and absolute file resource addresses under `workspace-path` for addressing files within agent sessions.

— Names API concepts but gives no endpoint or command example `dsh-v0.1.5-alpha.1`

05 **Playwright video input in GIF recording skill** IMPROVED

65

Adds support for encoding Playwright video (WebM) input — in addition to ordered screenshots — in the `encode_gif.py` GIF recording skill.

— Names the exact file and format newly supported `dsh-v0.1.5-alpha.1`

THINNER COVERAGE BELOW

06 **Unified feedback dialog** NEW

55

Opens a unified feedback dialog from `/feedback` and Dislike actions, with categories and a confirmation toast.

— Names the /feedback entry point but no further mechanism `dsh-v0.1.5-alpha.2`

07 **Reversible docking engine (dockkit) in client layout** NEW

55

Adds a reversible docking engine with pointer interactions (`dockkit`) and a responsive right column with width concessions to the layout.

— Names dockkit but gives no usage instructions `dsh-v0.1.5-alpha.1`

08 **CLI profile creation from shipped templates** NEW

50

Adds `create profiles from shipped templates` via the CLI, letting users bootstrap agent profiles from built-in starting points.

— Names the CLI capability but not the exact command syntax `dsh-v0.1.5-alpha.1`

09 **Stats UI rework with icon pills** IMPROVED 35

Replaces the stats strip with two icon pills and click-open stat dialogs in the client UI.

— Describes the UI change without further mechanism dsh-v0.1.5-alpha.1

10 **Weighted pull request approvals** NEW 30

Enforces weighted pull request approvals via a new approval-weight subsystem.

— No detail on weighting mechanism or configuration dsh-v0.1.5-alpha.2

11 **Download cards for file deliveries** NEW 30

Presents immutable file deliveries as download cards in the web UI.

— Bare description with no mechanism or location detail dsh-v0.1.5-alpha.2

12 **Parent-owned child catalogs in subagent subsystem** NEW 20

Records and observes parent-owned child catalogs in the subagent subsystem.

— Bare mention with no mechanism or usage given dsh-v0.1.5-alpha.2

WAS THIS USEFUL?  

AGENT

◆ AI Agent Frameworks

PydanticAI

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

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

> PydanticAI v2.42.0 adds a GitHub Copilot model provider and tightens validation of deferred tool approval results.

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

01 **GitHubCopilotProvider for Copilot API** NEW 50

Adds `GitHubCopilotProvider` to connect agents to GitHub Copilot's OpenAI-compatible API.

— Names the provider class but no usage example or config detail. v2.42.0

02 **Stricter validation of deferred tool approvals** BREAKING 50

Invalid `DeferredToolResults.approvals` values are now rejected, whereas previously they were accepted silently.

— Explains before/after behavior but no migration steps given. v2.42.0

└──▷ **BREAKING ON UPGRADE**

! Invalid `DeferredToolResults.approvals` values are now rejected (previously accepted silently).

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

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

> Vercel AI SDK 7.0.94 adds per-request model overrides for batch calls and refines retry behavior for empty image results.

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

01 **Smarter retry classification for empty image results** IMPROVED

55

Unclassified empty image results are now retried using provider-independent retryability classification, while Google and Google Vertex prompt blocks are marked terminal (non-retryable) so they no longer trigger useless retries.

— Explains mechanism and named providers but no code/flag to act on ai@7.0.94

02 **Per-request model overrides in batch calls** NEW

50

The `batch` call API now supports per-request model overrides, letting each item within a single batch target a different model instead of sharing one model for the whole batch.

— Names the surface (`batch`) but no example or config shown ai@7.0.94

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ 2 RELEASES · 2026-09-08

NOTES ↗ RANK

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

> Strands Agents shipped parallel Python (v1.55.0) and TypeScript (v1.17.0) releases that deepen MCP 2.x protocol support, add in-process agent-to-agent communication via a new LocalAgent protocol, introduce background task execution, and extend model caching to six providers, while dropping Node.js 20 support in favor of Node.js 22+.

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

01 **MCP server loading controls in load_servers** NEW 78

`load_servers` MCP calls now accept `prefix_with_server_name` and `continue_on_error` options, letting callers namespace tool names per originating server and tolerate a failed server load without aborting the whole session. Shipped identically in both the Python and TypeScript SDKs.

Prevent tool-name collisions when loading multiple MCP servers by prefixing each tool name with its originating server name.

typescript

```
loadServers([
  { name: 'search', url: 'http://localhost:3001' },
  { name: 'files',  url: 'http://localhost:3002' }
], { prefix_with_server_name: true, continue_on_error: true })
```

Load MCP servers with a name prefix and tolerant error handling so a missing server does not abort the whole session.

python

```
from strands.mcp import load_servers

tools = load_servers(
    servers=['filesystem', 'network-scanner'],
    prefix_with_server_name=True,
    continue_on_error=True
)
```

— Named flags with runnable examples in both SDKs typescript/v1.17.0 python/v1.55.0

02 **tools_ttl field in CacheConfig** IMPROVED 75

`CacheConfig` gained a `tools_ttl` field to set a time-to-live for cached tools, superseding the now-deprecated `cache_tools` field.

Set a TTL on cached tools so stale tool definitions are automatically invalidated without touching the broader cache strategy.

```typescript
const agent = new Agent({
  model: new BedrockModel({
    cacheConfig: {
      tools_ttl: 300
    }
  })
})
```

— Names both old and new fields with a runnable example `typescript/v1.17.0` `python/v1.55.0`

03 **MCP 2.x protocol support expansion** NEW 73

MCP 2.x support was expanded with `call_tool`, `server/discover` for SEP-2575 server discovery, MRTR (MCP Request/Response) extended to prompts and resources, MRTR input-required support, automatic tool refresh on list-changed notifications, and, in Python, SEP-2663 task support.

— Enumerates every named endpoint/spec but lacks examples `typescript/v1.17.0` `python/v1.55.0`

04 **cache_config support for new model providers** NEW 67

`cache_config` support was added to the Ollama, Writer, SageMaker, `llama.cpp`, Vercel, and Mistral model providers, extending provider-level tool/response caching to backends that previously had none.

Enable caching on a SageMaker-backed agent to reduce redundant inference costs on repeated tool definitions.

```python
from strands import Agent
from strands.models import SageMakerModel
from strands.models.cache import CacheConfig

model = SageMakerModel(
    endpoint_name='my-endpoint',
    cache_config=CacheConfig(tools_ttl=300)
)
agent = Agent(model=model)
agent('Summarize the threat report.')
```

— Names all six providers and the config key with an example `typescript/v1.17.0` `python/v1.55.0`

05 **web_fetch vended tool** NEW 60

A new `web_fetch` vended tool lets agents perform HTTP retrieval directly during a session.

Use the new web_fetch vended tool so an agent can retrieve live threat intel or CVE details during a session.

```python
from strands import Agent
from strands_tools import web_fetch

agent = Agent(tools=[web_fetch])
agent('Fetch https://nvd.nist.gov/feeds/json/cve/1.1/recent.json
and summarise the top 5 critical CVEs.')
```

— Named tool with a runnable usage example `python/v1.55.0`

THINNER COVERAGE BELOW

06 **Reconnect support for bidirectional streaming adapters**

IMPROVED 52

The OpenAI and Gemini Google Live bidirectional streaming model adapters gained a reconnect() method to recover dropped streaming sessions, and bidirectional sessions now support proactive reconnect ahead of drops.

— Names the method and two adapters, no example `typescript/v1.17.0` `python/v1.55.0`

07 **Background task execution in Python and TypeScript** NEW 52

Python agents gained an internal in-process task engine and manager, and the TypeScript `Agent` class gained a `backgroundTasks` option backed by a new `BackgroundTaskManager` interface and `InProcessTaskManager` implementation.

– Names classes and field but no usage shown `python/v1.55.0`

---

08 **Context manager offloading and stash support** NEW 48

The context manager gained an L1 stash for durable storage of offloaded content and session support for the L1 context manager (TypeScript); Python ported the offloading strategies and stash integration; context manager types are now exported as experimental.

– Names several mechanisms but no example or config key `typescript/v1.17.0` `python/v1.55.0`

---

09 **LocalAgent protocol for in-process agent-to-agent calls** NEW 45

A new `LocalAgent` protocol and implementation support in-process, bidirectional (bidi) streaming communication between agents, added to both the Python and TypeScript SDKs.

– Names the protocol but no usage example provided `typescript/v1.17.0` `python/v1.55.0`

---

10 **Classifier routing strategy in TypeScript** NEW 40

A classifier routing strategy was ported from the Python SDK to TypeScript, enabling model routing decisions based on request classification.

– Names the strategy but no mechanism or example given `typescript/v1.17.0`

---

11 **Node.js 22+ now required** BREAKING 40

Node.js 20 is no longer supported; the TypeScript SDK now requires Node.js 22+.

– Clear before/after but no migration steps given `typescript/v1.17.0` `python/v1.55.0`

---

12 **QmdSearchStrategy BM25 full-text search** NEW 38

A new `QmdSearchStrategy` adds BM25 full-text search to the storage layer.

– Names the class but no mechanism detail or example `python/v1.55.0`

---

13 **AudioCapable interface for bidi streaming** NEW 30

An `AudioCapable` interface and typed model config were added to the TypeScript bidirectional streaming layer.

– Bare name, no explanation of behaviour `typescript/v1.17.0`

---

14 **Bedrock API key authentication** NEW 25

Amazon Bedrock model provider now supports API key authentication.

— Single-line mention with no mechanism or example `python/v1.55.0`

---

15 **Multiple continuation inputs in Python agents** NEW

23

Python agents now support multiple continuation inputs.

— Thin one-line mention with no further detail `python/v1.55.0`

└──▷ **BREAKING ON UPGRADE**

! Node.js 20 is no longer supported; the TypeScript SDK now requires Node.js 22+.

! Node.js 20 is no longer supported; Node.js 22+ is now required.

WAS THIS USEFUL?  

Agno (formerly Phidata)

Feedback ⓘ 1 RELEASE · 2026-09-08

NOTES ↗ RANK

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

> Agno v3.0.8 adds bounded full-page reads to Knowledge, PostgreSQL engine factories with pool and serialization defaults, and a shared Markdown code-fence utility.

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

01 **Full-page reads for Knowledge** NEW 83

Adds `Knowledge.read_full_page` and `aread_full_page` methods that return an entire published page from a single bounded SQL read, supporting revision pinning, character limits, and deadlines; oversized pages return `None`.

— Named methods with clear mechanism and bounds, but no usage example. v3.0.8

02 **PostgreSQL engine factory functions** NEW 81

Adds `create_postgres_engine` and `create_async_postgres_engine` factory functions that expose Agno's connection-pool and JSON serialization defaults with configurable SQLAlchemy options; plain `postgres://` and `postgresql://` URLs automatically select Psycopg 3.

— Named functions and driver-selection behaviour, no runnable example given. v3.0.8

THINNER COVERAGE BELOW

03 **Shared Markdown code-fence tracking utility** NEW 58

Exposes `agno.utils.markdown.advance_code_fence` to share the page chunker's fence-tracking rules with application transforms that need to preserve code examples.

— Names the exact utility path but describes purpose only briefly. v3.0.8

WAS THIS USEFUL? 👍 👎

OpenClaw

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ CODE ↗ RANK

Personal AI assistant platform that connects messaging channels, tools, and model providers through a gateway.

> OpenClaw v2026.9.3 ships public session transcript sharing, a searchable meeting library, an optional Team Reports plugin, and default bounded recursive session spawning, alongside a wide set of CLI, config, and client platform additions plus several breaking SDK and Node version changes.

└──▷ WHAT SHIPPED · 27 FEATURES most completely described first ? what's the number?

01 Plugin SDK breaking API changes

BREAKING 90

The execution-policy SDK moves retired exec-mode and comparator helpers from `infra-runtime` to `execPolicy` on `openclaw/plugin-sdk/agent-harness-runtime`; callers must use `resolveExecModePolicy` and select only the returned fields they need. The approval SDK requires importing approval account-resolution helpers from `approval-native-runtime`, adapting session filtering to the full `matchesApprovalRequestFilters` contract, and replacing the retired generic forwarding evaluator with native channel route gates and shared predicates. SDK alias `channel-inbound.buildChannelTurnMediaPayload` is replaced by `buildChannelInboundMediaPayload`, and the named type `AbortAndDrainAgentHarnessRunResult` is removed — infer the result of the retained `abortAndDrainAgentHarnessRun` callable instead. Search result callbacks must read bounded Find/Grep text from `details.content` instead of the retired `details.truncation.content` (the outer tool-message `content` is unchanged), and directory result callbacks must use `LsToolDetails.content` and optional `nextAfter` in place of the retired `truncation` and `entryLimitReached` fields, passing `nextAfter` as `after` to continue listing.

— Names every retired and replacement API surface verbatim. v2026.9.3

02 Workshop skills migrated to per-agent collections

BREAKING 86

Workshop skills ownership moves from workspace-owned to one writable collection per agent; the config key `skills.workshop.allowSymlinkTargetWrites` is retired. Running `openclaw doctor --fix` migrates proven legacy skills automatically, leaving skills with ambiguous ownership in place for manual review.

Migrate legacy workspace-owned Workshop skills to the new per-agent collection after upgrading.

```
$ openclaw doctor --fix
```

— Names exact migration command and retired config key. v2026.9.3

03 **Node version support requirements raised** BREAKING 83

Node 22, Node 25, and earlier 24.x/26.x builds are no longer supported; upgrade to Node 24.16.0 or newer on the 24.x line, or Node 26.1.0 or newer, before upgrading OpenClaw, or SQLite text will be silently truncated. Node-based CLI/Gateway installs on macOS 11–13.4 or official Linux ARMv7 no longer have a supported host.

— Gives exact version thresholds and consequence of ignoring them. v2026.9.3

04 **`gateway.cliAgents.enabled` config key** NEW 81

New config key `gateway.cliAgents.enabled` (default `true`) controls whether supported CLI agents appear in the new-session model picker; set `gateway.cliAgents.enabled: false` to disable CLI agents and native CLI session creation.

Disable CLI agents on a shared Gateway where only UI-created sessions should be allowed.

```yaml
gateway.cliAgents.enabled: false
```

— Names exact config key, default, and effect. v2026.9.3

05 **Team Reports plugin** NEW 78

Optional Team Reports plugin generates authenticated daily, weekly, and monthly GitHub reports and explicitly selected Discord sources, with stored history, optional summaries, people timelines and calendars, coverage warnings, scheduler/source health, and theme-aware report pages.

— Rich feature list but no config keys or commands given. v2026.9.3

06 **Searchable meeting library** NEW 75

Adds a searchable meeting library: browse paginated saved notes, search full transcripts, export complete Markdown or JSONL archives, and edit capture sources in Communications settings.

— Names export formats and settings location. v2026.9.3

07 **Code Mode sandboxed encoding and TypeScript diagnostics** NEW 75

Code Mode adds sandboxed `TextEncoder` and `TextDecoder` instances that survive wait/resume for local text and byte transformations, TypeScript checks against available tools, a running JavaScript environment retained across fast tool replies, and error reporting with original-source locations and bounded console output.

— Names exact APIs and behaviours added to Code Mode. v2026.9.3

---

08 **`openclaw docs --limit` flag** NEW 73

Adds a `--limit` flag to `openclaw docs` to cap the number of displayed terminal or JSON documentation results returned by the CLI, e.g. `openclaw docs --limit 20`.

Limit CLI doc output to a manageable number of results when piping plugin documentation into another tool.

```
$ openclaw docs --limit 20
```

— Names exact flag and shows runnable command. v2026.9.3

---

09 **Typed MCP GetPromptResult from getPrompt** IMPROVED 65

`SessionMcpRuntime.getPrompt` now returns the typed MCP `GetPromptResult` shape, preserving messages and image content, and validates external responses instead of returning arbitrary unknown values.

— Names exact API and type, no usage example given. v2026.9.3

---

10 **Public session transcript sharing** NEW 65

Session owners and Gateway admins can publish existing and future conversation text to a public URL accessible to anyone, and revoke access later. The read-only public view omits tools, reasoning, files, images, and executable widgets.

— Clear scope and exclusions, no exact UI path given. v2026.9.3

---

11 **Runtime identity and GC duration metrics** NEW 65

Adds Prometheus runtime identity metrics so operators can verify which process and build is serving a deployment, plus garbage-collection duration metrics that identify slow phases while opening agent databases.

— Names Prometheus and metric purpose, no metric names given. v2026.9.3

---

12 **Repository-backed cloud sessions** NEW 63

Supports creating cloud sessions from a repository URL and ref, including checkout, setup, and recoverable checkpoints owned by the cloud node.

— Explains mechanism, no command or config key named. v2026.9.3

---

13 **Native dashboard reports via `show_widget`** NEW 62

Adds native dashboard report display through `show_widget` and dashboard authoring without an iframe.

— Names the exact `show_widget` surface used. v2026.9.3

14 **Provider account controls in Models settings** NEW 60

Models settings gains controls to add and remove individual provider accounts, manage supported account priority, and clear an agent's custom order without disconnecting accounts.

— Names UI area and controls, no exact command. v2026.9.3

15 **Live browser tab repainting for agent pages** NEW 60

Adds live browser tab repainting for agent pages, with screenshot fallback when streaming is unavailable; on macOS, external links open as native WebKit Mac tabs in the Browser panel that survive chat switches.

— Explains fallback and platform behaviour, no config surface. v2026.9.3

THINNER COVERAGE BELOW

16 **Model fallback picker** NEW 58

A model fallback picker lets users choose ordered fallback models from a searchable dropdown, remove selections, and add custom model references.

— Describes UI mechanism without exact navigation path. v2026.9.3

17 **Bounded update repair and consent-based failure reports** NEW 58

Candidate-validation failures during updates can enter a bounded repair phase using configured inference in disposable rehearsal state, with failure or rollback outcomes retained when repair cannot recover. Update failure reports are submitted only after explicit consent, using the recorded update attempt and current owner.

— Explains mechanism in prose, no exact command or flag. v2026.9.3

18 **Telegram photo album grouping** IMPROVED 55

Groups consecutive eligible Telegram photo attachments into native albums of up to ten while preserving order, captions, reply targets, and topics.

— Concrete limit of ten, but automatic behaviour with no user action. v2026.9.3

19 **Android folding layout support** NEW 53

Keeps content clear of hinges, shows navigation beside content on book folds, and splits transcript and composer across tabletop panes on Android foldables.

— Describes behaviour but no way to trigger manually. v2026.9.3

20 **ARM64 Linux AppImage and macOS Gateway cards** NEW 52

Adds ARM64 Linux AppImage packaging and live Gateway cards with direct window opening in the macOS Gateways menu.

— Names packaging format and menu location, thin on mechanism. v2026.9.3

21 **Bounded recursive session spawning** NEW 50

Recursive session spawning is enabled by default, bounded by explicit depth and concurrency limits, with existing sandbox restrictions retained.

— Describes limits generally but no concrete numbers or flags. v2026.9.3

22 **Chrome extension setup on Mac** NEW 50

Adds Chrome extension setup on Mac: request the official Chrome Store extension from local Mac device settings or CLI once native setup is ready.

— Names two entry points but no exact command. v2026.9.3

23 **Device aliasing in Control UI** NEW 45

Adds device aliasing: rename paired devices directly from the Control UI Devices page.

— Names exact UI page for the feature. v2026.9.3

24 **iOS 'Start Live Voice' shortcut** NEW 45

Adds an iOS voice shortcut: the Start Live Voice App Shortcut opens the current chat and starts Talk.

— Names the exact App Shortcut. v2026.9.3

25 **Live activity session status display** NEW 44

Shows running and queued permitted sessions in Live activity on entry and reconnect, with a visible limit notice and connection status.

— Describes UI state, no exact navigation given. v2026.9.3

26 **Position rail for long conversations** NEW 42

Adds a previewable position rail for long conversations, including hover previews and direct jumps to points in the conversation.

— Thin UI description, no exact location named. v2026.9.3

27 **Instant path filtering in folder browser** NEW 30

Adds instant path filtering in the folder browser as you type.

— Bare description of a small UI convenience. v2026.9.3

└──▷ **BREAKING ON UPGRADE**

- ! Node 22, Node 25, and earlier 24.x/26.x builds are no longer supported; upgrade to Node 24.16.0 or newer (on the 24.x line) or Node 26.1.0 or newer before upgrading OpenClaw, or SQLite text will be silently truncated. Node-based CLI/Gateway installs on macOS 11–13.4 or official Linux ARMv7 no longer have a supported host.
- ! Execution-policy SDK: the retired exec-mode and comparator helpers move from `infra-runtime` to `execPolicy` on `openclaw/plugin-sdk/agent-harness-runtime`; callers must use `resolveExecModePolicy` and select only the returned fields they need.
- ! Approval SDK: import approval account-resolution helpers from `approval-native-runtime`, adapt session filtering to the full `matchesApprovalRequestFilters` contract, and replace the retired generic forwarding evaluator with native channel route gates and shared predicates.
- ! SDK alias `channel-inbound.buildChannelTurnMediaPayload` is replaced by `buildChannelInboundMediaPayload`; the named type `AbortAndDrainAgentHarnessRunResult` is removed — infer the result of the retained `abortAndDrainAgentHarnessRun` callable instead of importing it.
- ! Search result callbacks: read bounded Find/Grep text from `details.content` instead of the retired `details.truncation.content`; the outer tool-message `content` is unchanged.
- ! Directory result callbacks: use `LsToolDetails.content` and optional `nextAfter` in place of the retired `truncation` and `entryLimitReached` fields; pass `nextAfter` as `after` to continue listing.
- ! Workshop skills: workspace ownership is replaced by one writable collection per agent; the config key `skills.workshop.allowSymlinkTargetWrites` is retired. Run `openclaw doctor --fix` to migrate proven legacy skills; ambiguous ownership is left in place for manual review.

WAS THIS USEFUL? 👍 👎

Mastra

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

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

> Mastra 1.65.0 expands observability with tenant-scoped batch trace deletion and a recursive trace query API, adds interactive agent-channel action handling and message-author tagging, and ships several breaking changes to @mastra/factory and @mastra/playground-ui.

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

01 **Advanced trace query API** NEW 86

`traceQueryRequestSchema` (from `@mastra/core`) supports bounded time ranges, recursive predicates over traces/spans/scores, thread grouping, and deterministic cursor pagination; invalid or overly complex requests are rejected before a storage adapter executes them.

Query traces filtered by environment and time range — useful for isolating production issues in a specific window without loading all trace data.

typescript

```
import { traceQueryRequestSchema } from '@mastra/core';

const request = traceQueryRequestSchema.parse({
  timeRange: { from: '2026-08-01T00:00:00Z', to: '2026-09-
01T00:00:00Z' },
  where: { op: 'eq', left: { path: 'environment' }, right: {
literal: 'production' } },
});
```

— Runnable schema example with clear filtering/pagination mechanism `@mastra/core@1.65.0`

02 **Agent channel action handlers** NEW 80

`handlers.onAction` in agent channel configuration lets apps handle button clicks and select changes from interactive UI cards; the built-in approval handler is exposed as `defaultHandler`, and setting `onAction: false` disables it.

Handle a `retry` button click from an agent channel card while still delegating other actions to Mastra's built-in approval handler.

```typescript
import { Agent } from '@mastra/core';
import { createSlackAdapter } from '@mastra/slack';

const agent = new Agent({
  channels: {
    adapters: { slack: createSlackAdapter() },
    handlers: {
      onAction: async (event, defaultHandler) => {
        if (event.actionId === 'retry') {
          await event.thread?.post('Retrying...');
          return;
        }
        await defaultHandler();
      },
    },
  },
});
```

— Runnable example shows delegating to defaultHandler and disabling `@mastra/core@1.65.0`

03 **Message author tagging via request context** NEW 77

`MASTRA_MESSAGE_AUTHOR_KEY` (from `@mastra/core/request-context`) tags every message sent in an agent-controller session with a sender identity object `{ id, name?, avatarUrl? }`, stored under `providerMetadata.mastra.author`.

Tag messages with the authenticated user's identity so a shared thread shows who wrote each message.

```typescript
import { MASTRA_MESSAGE_AUTHOR_KEY } from '@mastra/core/request-context';

// In your auth middleware, before the agent-controller session:
requestContext.setRaw(MASTRA_MESSAGE_AUTHOR_KEY, {
  id: user.id,
  name: user.name,
  avatarUrl: user.avatarUrl,
});
```

— Exact key, storage location and runnable middleware example `@mastra/core@1.65.0`

04 **Tenant-scoped batch trace deletion** NEW 72

storage.batchDeleteTraces({ traceIds, organizationId }) deletes up to 1,000 trace IDs per batch, cascading cleanup across spans and trace-linked scores, feedback, metrics, and logs.

— Named method and params but no runnable example `@mastra/core@1.65.0`

05 **`@mastra/factory` board and API breaking changes** BREAKING

72

defineBoard() phases must now include a `kind` field, working phases must include a `role` field, and `initialPhase` must be `resting`. The global rules object is removed — `FactoryRules`, its defaults/merge/assert helpers, and new MastraFactory({ rules }) are gone; tool-result rules now go on defineBoard({ tools }) and `configVersion` replaces `rules.version`. The `GET /web/factory/projects/:id/attention` response shape changed: the `tier` field is removed and counts/latest fields moved under `kinds[kind]...`.

— Names every removed/changed field, method, and endpoint for migration `@mastra/core@1.65.0`

06 **Metadata fields on workflow control-flow entries** IMPROVED

70

Optional `id`, `description`, and `metadata` are now supported on workflow control-flow entries .parallel(), .branch(), .dowhile(), .dountil(), .foreach(), .sleep(), .sleepUntil(), and .map(); these fields survive serialization and rehydration and appear in `serializedStepGraph`.

— Lists every method and field but lacks a usage example `@mastra/core@1.65.0`

07 **Span tracing API and new built-in span types** IMPROVED

67

Processors can now declare `spanType`, `spanName`, and `spanAttributes` (using `SpanType` from `@mastra/core/observability`) to control how they appear in traces. New built-in span types `SKILL_ACTION` (covering the full skill lifecycle: resolve, inject, activate, search, read) and `AGENT_SIGNAL` (recording each state signal emission) were added, and built-in subsystems now emit named spans such as `skill:inject`, `workspace:mount:instructions`, and `memory: recall` instead of anonymous processor-phase names.

— Names schema fields and span types but no code example `@mastra/core@1.65.0`

08 **Experiment rename and deletion in datasets** NEW

62

dataset.updateExperiment({ experimentId, name, description }) renames an experiment or updates its description and metadata after creation, and mastra.datasets.deleteExperiment({ experimentId }) deletes an experiment along with

its traces, spans, scores, feedback, metrics, and logs, even when orphaned from its dataset.

— Two named methods with params, no usage example `@mastra/core@1.65.0`

THINNER COVERAGE BELOW

09 `@mastra/playground-ui` trace panel breaking changes

BREAKING

58

`TraceDataPanelView` replaces `partialThreadTabSlot` with `messagesPanelSlot`, and `TracesLayout` replaces `sidePanelWide` with `sidePanelWidth`, which accepts `'half' | 'wide' | 'full'`.

— Exact prop renames given but no migration example `@mastra/core@1.65.0`

└──▷ BREAKING ON UPGRADE

! `@mastra/factory`: defineBoard() phases must now include a `kind` field, working phases must include a `role` field, and `initialPhase` must be `resting`.

! `@mastra/factory`: The global rules object is removed — `FactoryRules`, defaults/merge/assert helpers, and new MastraFactory({ rules }) are gone; tool-result rules move onto defineBoard({ tools }) and `configVersion` replaces `rules.version`.

! `@mastra/factory`: `GET /web/factory/projects/:id/attention` response shape changed — the `tier` field is removed and counts/latest fields have moved under `kinds[kind]...`.

! `@mastra/playground-ui`: `TraceDataPanelView` replaces `partialThreadTabSlot` with `messagesPanelSlot`; `TracesLayout` replaces `sidePanelWide` with `sidePanelWidth` accepting `'half' | 'wide' | 'full'`.

WAS THIS USEFUL? 👍 👎

OpenAI Agents SDK

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

RANK

OpenAI Agents SDK is an open-source framework that provides tools, handoffs, guardrails, and tracing for agentic applications.

> OpenAI Agents SDK v0.22.2 adds support for current image generation tool options when configuring agents.

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

01 **Current image generation tool options support** IMPROVED

26

The SDK now supports current image generation tool options when configuring agents, though specific option names or parameters are not detailed in the release notes.

— No specific flags, params, or mechanism named v0.22.2

WAS THIS USEFUL? 👍 👎

Mem0

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

Mem0 is a memory layer for AI agents that extracts, stores, and retrieves durable facts across sessions through a hosted or self-hosted store.

> Mem0's DeepSeek plugin now automatically recalls and captures memories across a session, ships as a standalone ESM package, and locks down cross-user memory access behind an explicit operator flag.

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

01 **Automatic memory recall and capture** NEW 80

The DeepSeek plugin adds `autoRecall` and `autoCapture` boolean options (both default `true`) that control automatic memory recall during `system-prompt/assemble` and automatic capture from the durable `session/event` stream after a completed turn. Recall uses the latest human prompt and skips repeated context injection within the same session, while capture is skipped for interrupted or incomplete turns.

— Named config flags and stream mechanism fully explained deepseek-plugin-v0.3.0

02 **Operator opt-in for cross-user memory overrides** BREAKING 70

Cross-user `userId` overrides on the `search_memory` and `add_memory` tools now require explicit operator opt-in via `allowUserOverride: true`; existing setups relying on unrestricted `userId` overrides will stop working without this flag.

— Exact flag and affected tools named, migration implied deepseek-plugin-v0.3.0

THINNER COVERAGE BELOW

03 **Standalone @mem0/deepseek-plugin ESM package** NEW 45

Publishes a self-contained ESM artifact under `@mem0/deepseek-plugin`, bundling native Harness services while keeping the Mem0 SDK as an external dependency.

— Names the package but no usage detail deepseek-plugin-v0.3.0

└──▷ **BREAKING ON UPGRADE**

! Cross-user `userId` overrides on `search_memory` and `add_memory` now require explicit operator opt-in via `allowUserOverride: true`; existing setups relying on unrestricted `userId` overrides will stop working without this flag.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

emisar

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

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

> Emisar v0.47.0 adds new Stripe and Braintree packs and extends the Airflow pack to cover both Airflow 2 and Airflow 3, alongside a round of access-control changes for Owners, SSO groups, and shared reads plus runner and console reliability fixes.

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

01 **Airflow 2 and 3 support in Airflow pack** IMPROVED 70

The Airflow pack now supports Airflow 2 (API v1) and Airflow 3 (API v2), with jobs, assets, and backfills requiring Airflow 3.

— States exact API versions and which features require Airflow 3 v0.47.0

02 **Runner setup key 24-hour expiration** BREAKING 65

Existing unused console-generated runner setup keys are assigned a 24-hour lifetime; any old expired key must be replaced with a new setup command.

— Gives exact lifetime and required remediation action v0.47.0

03 **SSO group role and access management on provider page** IMPROVED 60

SSO groups now show role, runner, and pack access together on the provider page, with in-place role editing and member filtering.

— Names a UI location and specific controls added v0.47.0

04 **Approval recovery error differentiation** IMPROVED 60

Approval recovery distinguishes catalog failures, unavailable actions, and changed contracts, preserving entered notes, inputs, and unsaved runbook edits during runner/pack access changes.

— Details multiple failure cases and what state is preserved v0.47.0

THINNER COVERAGE BELOW

05 **Stripe and Braintree billing packs** NEW 50

Adds new Stripe and Braintree packs covering billing investigation, refunds, disputes, and related corrections.

— Names the two packs and use cases but no mechanism v0.47.0

06 **Owner-wide action access normalization** BREAKING 50

Owners always have workspace-wide action access; existing Owner scopes are normalized with an audit record. This normalization is audited and does not restore narrower grants on rollback.

— Explains behavior and rollback caveat but no config surface v0.47.0

07 **Windows Hermes and Goose setup paths** IMPROVED 40

Windows Hermes and Goose setup uses native configuration paths, with setup guidance now including co:op.

— Names tools and a new guidance addition, no detail on mechanism v0.47.0

08 **Runner credential rotation and dispatch journal durability** IMPROVED 40

Runner credential rotation uses the authenticated connection, and the dispatch journal appends durable transitions with periodic compaction.

— Describes mechanism but no exposed interface v0.47.0

09 **Console runbook plan review retention** IMPROVED 40

Console runbook starts retain the release and plan reviewed; changed plans require another review.

— Describes behavior change with modest specificity v0.47.0

10 **Shared reads no longer authorize actions** BREAKING 35

Shared reads no longer authorize execution, cancellation, approval, or access changes.

— Clear restriction but no migration path given v0.47.0

11 **Operational role workspace-wide activity visibility** NEW 30

Operational roles can inspect workspace-wide activity while actions remain limited by current runner and pack access.

— Single sentence with no mechanism or surface named v0.47.0

12 **Credential-aware GCP project discovery** IMPROVED 25

GCP project discovery follows the current credentials.

— One-line change with no further detail v0.47.0

└──▷ BREAKING ON UPGRADE

- ! Existing unused console-generated runner setup keys are assigned a 24-hour lifetime; any old expired key must be replaced with a new setup command.
- ! Owner access normalization is audited and does not restore narrower grants on rollback.
- ! Rolling back the new group-access constraints can fail if newer independent grants still depend on them.

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ

2 RELEASES · 2026-09-08 → 2026-09-09 NOTES ↗ RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> This window's OpenAI TypeScript SDK updates added GPT Image 2.5 model support with new image options, plus API-level additions for prompt cache diagnostics, service-account key expiration, and web search call status handling.

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

01 **Incomplete status for web search calls** IMPROVED 45

Recognizes `incomplete` as a valid web search call status in streaming and response types.

— Names the exact status value and affected types. v7.11.0

02 **GPT Image 2.5 model support** NEW 35

Adds support for GPT Image 2.5 models along with new image options in the API.

— Names the model but not the new options or usage. v7.12.1

03 **Prompt cache diagnostics in responses** NEW 30

Adds prompt cache diagnostics to API responses, exposing cache performance data.

— Describes what was added but not the response fields or format. v7.11.0

04 **Service-account API key expiration fields** NEW 25

Adds service-account API key expiration fields to the SDK.

— Only mentioned in summary, no field names or mechanism given. v7.11.0

WAS THIS USEFUL? 👍 👎

XGrammar

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

XGrammar constrains language-model output to grammars and schemas with optimized token-mask generation.

> XGrammar v0.2.6 adds new grammar constructs and format controls, a cached Lark compilation API, expanded structural tag support for more model families, and new backend/platform coverage including an NPU backend and Windows ARM64 wheels.

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

01 Token and character limits for text/tokens formats NEW

81

Adds `max_tokens` and `max_chars` support for `AnyTextFormat` and `AnyTokensFormat`, letting callers cap structured generation output length directly via these format parameters without post-processing.

Cap structured generation output by token count using the new `max_tokens` parameter on `AnyTokensFormat` to enforce a hard budget without post-processing.

python

```python
import xgrammar as xgr

fmt = xgr.AnyTokensFormat(max_tokens=128)
```

— Named parameters and formats with a runnable example v0.2.6

02 Cached compile_lark API for Lark grammars NEW

75

XGrammar adds a cached `GrammarCompiler.compile_lark` API so that compiling the same Lark grammar across requests reuses the cached result instead of recompiling, reducing latency in high-throughput serving.

Use the cached `compile_lark` API to avoid recompiling the same Lark grammar on every request, reducing latency in high-throughput serving.

python

```python
import xgrammar as xgr

compiler = xgr.GrammarCompiler(tokenizer_info)
grammar = compiler.compile_lark(lark_grammar_str)
```

— Named API with runnable example, but caching mechanism only briefly described v0.2.6

THINNER COVERAGE BELOW

03 **Lark regex flags and structured substring expressions** NEW

33

XGrammar adds Lark regex flags and structured substring expressions to grammar definitions, expanding the expressiveness of Lark-based grammar syntax.

— Named additions but no mechanism or usage shown v0.2.6

04 **Structural tag support for Cohere XML, Kimi K3, EXAONE 4.0** NEW

33

Adds structural tag support for Cohere XML, Kimi K3, and EXAONE 4.0 model formats.

— Names three supported formats but no detail on mechanism or usage v0.2.6

05 **Windows ARM64 wheel support** NEW

28

Adds Windows ARM64 wheel support for XGrammar.

— Simple platform packaging addition, thinly described v0.2.6

06 **NPU token-bitmask backend** NEW

23

Adds an NPU token-bitmask backend to XGrammar.

— Bare mention of a new backend with no further detail v0.2.6

WAS THIS USEFUL?  

OpenAI Python SDK

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

RANK

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

> OpenAI Python SDK v3.10.0 adds support for GPT Image 2.5 models with new image options and introduces service-account API key expiration fields for programmatic key lifetime management.

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

01 **Service-account API key expiration fields** NEW 45

Adds service-account API key expiration fields to the API, enabling programmatic inspection and management of key lifetimes.

— Explains purpose but no field names or endpoint given v3.10.0

02 **GPT Image 2.5 model support** NEW 30

The SDK adds support for new GPT Image 2.5 models, including new image options.

— Named only in summary, no options or mechanism detailed v3.10.0

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ 1 RELEASE · 2026-09-08

NOTES ↗ RANK

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

> HyperFrames v0.8.32 expands its linting and validation checks to catch more animation and rendering pitfalls before they reach production.

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

01 New lint and check rules for animation pitfalls NEW

47

Adds lint rules flagging `autoAlpha` on clip elements, hidden-style opacity guards, undefined GSAP color variables, and repeated `fromTo` state leaks, plus a new Check that detects connectors pointing at nothing and stylesheets that leak into the frame.

— Names each rule but no usage or config detail given v0.8.32

WAS THIS USEFUL? 👍 👎

Fireworks AI

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

Fireworks AI provides hosted inference, model fine-tuning, and deployment APIs for open-weight models.

> Fireworks AI shipped a new US-only serverless endpoint for GLM 5.3 Flash, deployment tags with firectl CLI support, per-user spending limits, bring-your-own provider keys, custom SSO/SCIM provisioning, and cost estimation tools — alongside a breaking change enforcing a `custom/` namespace on deployment annotation keys.

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

01 Annotation keys now require custom/ namespace

BREAKING

80

The REST API now restricts customer-managed annotations to a `custom/` namespace; writes using bare keys (e.g. `environment`) return HTTP 403 (`PERMISSION_DENIED`) and must be migrated to a prefixed form (e.g. `custom/environment`). `GetDeployment` and `ListDeployments` now return only `custom/*` entries for regular account users, so existing reads relying on bare keys will silently return no data for those keys.

— Full before/after and named endpoints, but no migration command given

Deployment tags and annotation API chan…

02 GLM 5.3 Flash US-only serverless endpoint NEW

75

Adds `accounts/fireworks/routers/glm-5p3-flash-us` as a new US-only serverless routing endpoint for the GLM 5.3 Flash model, intended for data-residency compliance use cases.

Route inference requests to GLM 5.3 Flash via the US-only serverless endpoint for data-residency compliance.

```
$ curl -X POST
https://api.fireworks.ai/inference/v1/chat/completions \
  -H 'Authorization: Bearer <API_KEY>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "accounts/fireworks/routers/glm-5p3-flash-us",
"messages": [{"role": "user", "content": "Hello"}]}'
```

— Names exact model route and includes a runnable curl example product docs

03 Deployment Tags with firectl CLI support NEW

75

Adds Deployment Tags to attach customer-defined metadata to dedicated deployments, manageable via the UI, plus `firectl` 1.8.3 subcommands `deployment tag set`, `unset`, and `list`, including atomic batch operations for managing tags.

Attach metadata to a dedicated deployment for cost attribution or environment tagging.

> 📍 Navigate to Deployments, open a dedicated deployment, and add customer-defined key-value pairs under Deployment Tags.

— Names CLI version and subcommands but lacks full syntax detail

Deployment tags and annotation API chan…

---

04 **Per-user spending limits via Fireworks Nexus** NEW 65

Adds per-user spending limits on serverless inference through Fireworks Nexus, supporting account-level defaults, group limits, and per-user overrides configured under Usage Limits.

Enforce a spending cap for a specific end-user on serverless inference to prevent runaway costs in a multi-tenant application.

> 📍 Navigate to Fireworks Nexus › Usage Limits, set an account-level default, then override for an individual user or group under Per-User Usage Limits.

— UI navigation path given but no API or config surface named product docs

---

05 **Bring-your-own provider keys for FireRouter** NEW 65

Adds Provider Keys (Bring Your Own) support so FireRouter can call external providers using customer-supplied API keys, registered via Fireworks Nexus > Provider Keys.

Route inference requests through your own API keys for external providers via FireRouter without exposing them to end users.

> 📍 Navigate to Fireworks Nexus › Provider Keys, select Bring Your Own, and enter your provider API key to register it for FireRouter use.

— UI navigation path given, mechanism only briefly described product docs

THINNER COVERAGE BELOW

---

06 **Custom SSO and SCIM provisioning** NEW 40

Adds Custom SSO authentication and SCIM user and group provisioning for Fireworks AI accounts.

— Bare description with no setup steps or config named product docs

---

07 **Cost estimation tools for training and RL rollout** NEW

Adds a training cost estimator for Managed Training, comparing costs across Fireworks Serverless, Dedicated, and Tinker, and a separate RL rollout cost comparison tool for multi-turn agentic RL inference costs between Fireworks and Tinker.

— Names comparison targets but no usage path or numbers given `product docs`

└──▷ BREAKING ON UPGRADE

! REST API writes using bare annotation keys (e.g. `environment`) now return HTTP 403 (`PERMISSION_DENIED`); keys must be migrated to the `custom/` prefix (e.g. `custom/environment`).

! `GetDeployment` and `ListDeployments` now omit any annotation keys outside the `custom/*` namespace for regular account users — existing reads relying on bare keys will silently return no data for those keys.

WAS THIS USEFUL?  

Hugging Face Accelerate

Feedback ⓘ 1 RELEASE · 2026-09-09

NOTES ↗ RANK

Accelerate launches and runs PyTorch training across CPUs, GPUs, and TPUs with automatic mixed precision and configurable FSDP or DeepSpeed backends.

> Accelerate v1.15.0 adds CPU offload for FSDP2 activation checkpointing and a pluggable custom-tracker API, alongside a Neuron dynamo backend, FSDP2 tied-embedding support, per-rank sharded checkpointing, MPS accelerator handling, full disk offloading, and DTensor gradient clipping.

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

01 **Activation checkpointing offload in FSDP2** NEW 90

Adds `fsdp_activation_checkpointing_offload: true` to `fsdp_config` in FSDP2 YAML configs, moving checkpointed layer inputs to pinned CPU memory to cut activation memory at long sequence lengths.

Reduce activation memory during long-sequence FSDP2 training by offloading checkpointed layer inputs to pinned CPU RAM.

yaml

```yaml
# fsdp2.yaml
fsdp_config:
  fsdp_version: 2
  fsdp_auto_wrap_policy: TRANSFORMER_BASED_WRAP
  fsdp_activation_checkpointing: true
  fsdp_activation_checkpointing_offload: true
```

— names exact config key with example config file v1.15.0

02 **Custom experiment tracker registration** NEW 80

Adds `register_tracker_class` to allow custom trackers to be registered by name and selected via `log_with=` in `Accelerator`, the same as any built-in tracker.

Plug in a custom experiment tracker and select it by name, without patching Accelerate internals.

```python
from accelerate import Accelerator
from accelerate.tracking import register_tracker_class

register_tracker_class(MyTracker)  # MyTracker.name ==
'my_tracker'
accelerator = Accelerator(log_with='my_tracker')
```

— named API with runnable code sample v1.15.0

03 **Per-rank checkpoint save for FSDP2 sharded state** IMPROVED

60

Adds per-rank `torch.save` / `load` for `SHARDED_STATE_DICT` in FSDP2, resolving checkpoint timeouts on large NPU clusters.

— names API calls and fixed bottleneck, no example v1.15.0

THINNER COVERAGE BELOW

04 **Tied embedding support in FSDP2** NEW

55

Adds tied-embedding support for FSDP2 models on torch >= 2.13 by placing the output embedding in the same `fully_shard` group.

— names mechanism and version, lacks runnable example v1.15.0

05 **Gradient clipping support for DTensors** NEW

55

Adds gradient clipping support for DTensors, enabling `clip_grad_norm` across mixed plain-tensor and DTensor setups.

— names function and scope, no runnable example v1.15.0

06 **Neuron backend for torch dynamo** NEW

50

Adds a `neuron` backend for torch dynamo, enabling `--torch-compile` with the Transformers Trainer on Neuron devices.

— names flag and backend but no runnable example v1.15.0

07 **MPS accelerator reporting and OOM handling** IMPROVED

45

Reports MPS as the accelerator in `accelerate env` and handles MPS out-of-memory errors as OOM in `find_executable_batch_size`.

— names two commands, no example given v1.15.0

08 **Full disk offloading for entire models** NEW

30

Supports full disk offloading for entire models including tied weights, enabling compression workflows on memory-constrained machines.

— thin prose, no named flag or command v1.15.0

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

MLX Serve

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

The mlx-serve runtime provides Apple Silicon inference with OpenAI- and Anthropic-compatible APIs, chat, agent mode, and tool calling.

> mlx-serve v26.9.2 adds per-model configuration, multi-provider model registration, disk-backed prefix caching, and new speculative-decoding controls, alongside faster constrained JSON decoding and a chat UI width toggle.

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

01 Multi-provider model registration NEW

83

Adds `~/.mlx-serve/providers.json` for configuring OpenAI-compatible external providers (cloud APIs, remote Macs, local runtimes), whose models appear in the picker as `<model>@<name>`; the same registration is available via Settings > Providers UI to register any OpenAI-compatible chat server with its key.

— Names config file, naming convention, and UI location v26.9.2

02 Disk-backed prefix cache spill NEW

80

Adds `--prefix-cache-disk` flag to spill inactive conversation prefix caches to SSD, keeping the active model and conversation in memory while restoring others in seconds.

Run a long-context Flash Next server where inactive conversations spill to disk and speculation is cut off past 32k tokens to avoid diminishing returns.

```
$ mlx-serve launch flash-next --prefix-cache-disk --max-mtp-ctx
32768
```

— Names flag and mechanism, no size limits given v26.9.2

03 Per-model settings for context, cache, speculation NEW

80

Adds `~/.mlx-serve/model-settings.json` for per-model configuration of context size, KV cache precision, and speculative-decoding defaults, applied at load or hot-applied to a running model; the same settings are editable via right-click on a model in My Models > Model Settings.

— Names config file and exact UI path for all fields v26.9.2

04 Default reply token budget flag NEW

70

Adds `--max-tokens N` serve-mode flag to set a default reply token budget for clients that omit `max_tokens`.

— Exact runnable flag but no default value stated v26.9.2

05 **Speculative decoding context cutoff** NEW 65

Adds `--max-mtp-ctx <n>` flag to set a hard context-length cutoff beyond which speculative decoding is disabled.

Run a long-context Flash Next server where inactive conversations spill to disk and speculation is cut off past 32k tokens to avoid diminishing returns.

```
$ mlx-serve launch flash-next --prefix-cache-disk --max-mtp-ctx 32768
```

— Exact runnable flag but no default or threshold guidance v26.9.2

THINNER COVERAGE BELOW

06 **Full-speed constrained JSON decoding on Flash Next** IMPROVED 55

Delivers constrained (JSON schema) decoding at full model speed on Flash Next, replacing a previous ~1 token/second cap.

— Gives before/after numbers but no invocation steps v26.9.2

07 **Adaptive speculative decoding on Flash Next** NEW 50

Adds adaptive speculative decoding that automatically disables and re-enables speculation on Flash Next based on per-step cost/benefit measurement.

— Explains mechanism but nothing for reader to invoke v26.9.2

08 **Chat UI width control** NEW 50

Adds chat UI width control via Narrow/Medium/Wide settings, accessible from Settings or F1–F3 keyboard shortcuts.

— Named shortcuts, but a minor cosmetic addition v26.9.2

└──▷ ALSO FROM THESE RELEASES

WAS THIS USEFUL? 👍 👎

vMLX

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

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

> vMLX 1.6.56 adds native MiniCPM5 tool/reasoning support, expands per-request image/video controls with new diagnostics, introduces atomic safetensors alignment repair during model preflight, and redesigns the Console Amber UI alongside caching and parsing fixes.

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

01 **Safetensors alignment repair during model preflight** NEW 70

Adds atomic on-disk `safetensors` alignment repair during local model preflight, including nested draft/media shards, with validation and unsafe-path protections.

— Detailed mechanism and scope, but automatic with no command v1.6.56

02 **Console Amber UI redesign and settings grouping** IMPROVED 70

Adds new Chat & Images, Servers & API, and Models navigation sections in the Console Amber UI with an updated app icon and General Preferences; server settings now group concurrent processing, SSD prefix caching, power, generation, MCP, and speculative decoding controls together.

— Names every navigation section and setting group, UI path given v1.6.56

THINNER COVERAGE BELOW

03 **MiniCPM5 native tool/reasoning integration** NEW 55

Adds MiniCPM5 native tool/reasoning integration with guarded affine projection handling.

— Names mechanism but no usage path given v1.6.56

04 **Per-request image/video controls and diagnostics** NEW 55

Adds per-request image/video controls and typed strict-mode errors across API dialects, plus effective-setting diagnostics for image/video API requests.

— Describes behaviour but no concrete flag or endpoint v1.6.56

05 **Hybrid cache corrections and SSD cleanup** IMPROVED 50

Adds hybrid partial-prefix and media-position cache corrections, idle SSD cleanup, and capacity-only notices.

— Names cache mechanisms but no configuration surface v1.6.56

06 **Tool/JSON parsing and Ollama streaming fixes** IMPROVED

40

Tool/JSON parsing now preserves string data and terminal punctuation; Ollama generation streams retain final usage without repeating content.

— Describes two fixes but no reproduction steps v1.6.56

07 **JANG model feed filtering** IMPROVED

25

JANG model feeds now filter out non-JANG repositories.

— Bare statement with no mechanism or scope v1.6.56

WAS THIS USEFUL?

vLLM

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

The vLLM inference engine runs large language models with high-throughput, memory-efficient GPU execution.

> vLLM's v0.29.0 release adds request-queue admission control, a sharded P2P backend for RL weight sync, per-request speculative-decoding metrics, and support for five new model families, while a new docs example ships a ready-made Prometheus/Grafana observability stack for the OpenAI-compatible server.

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

01 Prometheus/Grafana observability example NEW 90

The OpenAI-compatible server exposes raw Prometheus metrics at `http://localhost:8000/metrics` by default with no extra configuration. A new `docker-compose.yaml` launches Prometheus (port `9090`) and Grafana (port `3000`) wired via Docker DNS, plus a `grafana.json` dashboard importable at `http://localhost:3000/dashboard/import` visualising `vllm:e2e_request_latency_seconds`, `vllm:inter_token_latency_seconds`, `vllm:prompt_tokens_total`, and `vllm:generation_tokens_total` with P50/P90/P95/P99 panels and a token-throughput timeseries.

Spin up the full Prometheus + Grafana monitoring stack alongside a running vLLM server to get live latency and throughput dashboards.

```
$ vllm serve mistralai/Mistral-7B-v0.1 --max-model-len 4096 &
docker compose up
```

Import the pre-built vLLM Grafana dashboard after connecting Prometheus as a data source at `http://prometheus:9090`.

> 📍 Navigate to `http://localhost:3000/dashboard/import`, upload grafana.json, and select the prometheus datasource.

— Runnable commands, exact ports, endpoint and metric names given product docs

02 Request-queue admission control flags NEW 75

`--max-num-queued-reqs` and `--max-num-queued-tokens` CLI flags cap queued load on the request queue to guard against runaway growth during traffic spikes.

Cap queued load during traffic spikes to prevent runaway queue growth and enforce admission control.

```
$ vllm serve meta-llama/Llama-3.1-8B-Instruct --max-num-queued-reqs 500 --max-num-queued-tokens 131072
```

— Two named flags plus a runnable serve command v0.29.0

03 **FlashInfer backend enhancements** IMPROVED 75

`--linear-backend flashinfer_cutedsl` opt-in flag enables FlashInfer CuTeDSL BF16 low-latency GEMM, and `VLLM_ALLREDUCE_USE_FLASHINFER=0` environment variable lets users opt out of FlashInfer all-reduce, which is now enabled by default for TP CUDA groups.

— Two named flags with default behaviour change described v0.29.0

04 **RL weight sync backends** NEW 75

A new `sharded_rdt` P2P backend for RL weight sync lets each worker pull only its TP/EP slice over NIXL or Ray Direct Transport, alongside rank-local IPC weight updates for RL workflows.

— Mechanism and named transports given, but no example v0.29.0

05 **New model family support** NEW 65

Adds support for Hy4-preview (Tencent 770B/49B-active MoE with Gated DeepSeek Sparse Attention and native MTP), Qwen3.8-Flash-Next (BF16/FP8/NVFP4 with MTP), GraniteSWA and GraniteMoeSWA, NemotronH_Omni_Reasoning_V3 with MTP, and Kimi K3 NVFP4 checkpoints.

— Five named models with architecture details, no usage steps v0.29.0

06 **Legacy architecture and backend cleanup** BREAKING 65

Ten deprecated model architectures are removed; FlexOlmo, Olmo3, and Hunyuan V1/VL are migrated to the Transformers modeling backend, removing their native implementations; the PyAV video decoder backend is removed; and the `VLLM_TEST_FORCE_FP8_MARLIN` environment variable is removed.

— Named removals and migrations give clear upgrade impact v0.29.0

07 **Model Runner V2 default and capability expansion** IMPROVED 60

Model Runner V2 is now the default for all models, completing the rollout begun with pooling models, and gains `extract_hidden_states` speculation support and prompt embeds support.

— Rollout scope and two capability additions named v0.29.0

THINNER COVERAGE BELOW

08 **Per-request speculative decoding metrics** NEW 55

`--per-request-spec-decode-metrics` CLI flag exposes per-request speculative decoding acceptance stats in OpenAI API responses.

— Named flag but no example or acceptance-stat detail v0.29.0

09 **Anthropic Messages API render endpoint** NEW 55

Adds `/v1/messages/render` endpoint for the Anthropic Messages API.

— Named endpoint path but no method or usage shown v0.29.0

10 **Deterministic prefix-cache hashing** IMPROVED 55

Prefix-cache `NONE_HASH` is now deterministic by default, removing the need to pin `PYTHONHASHSEED` for distributed KV cache users.

— Before/after behaviour named, no example command v0.29.0

11 **Multimodal input handling improvements** IMPROVED 55

The Python frontend now accepts video embeds, modality-scoped `mm_processor_kwargs` are honored for multimodal inputs, and ViT full CUDA graph support is added for Idefics3 and SmolVLM.

— Three named multimodal additions, no example commands v0.29.0

12 **OpenAI API server entrypoint deprecated** DEPRECATED 55

`python -m vllm.entrypoints.openai.api_server` is deprecated in favor of `vllm serve`.

— Exact old and new commands given for migration v0.29.0

13 **Mamba prefix cache retention control** NEW 50

`prefix_cache_retention_interval` CLI argument (default `0`) controls Mamba prefix cache retention.

— Named flag and default value, no further mechanism v0.29.0

14 **CPU offload for vision and audio towers** IMPROVED 50

`--cpu-offload-params` support is extended to vision and audio towers.

— Named flag, scope of extension stated, no example v0.29.0

15 **LoRA support expansion** NEW 50

Adds LoRA support for DeepSeek V4, Qwen3-Omni multimodal, and tower/connector LoRA for LLaVA-NeXT and LFM2-VL.

— Named models but no mechanism or usage detail v0.29.0

16 **Embedding model support additions** NEW 45

Adds bidirectional attention for DeepSeek-backbone embedding models and FP8 quantization support for ModernBERT.

— Two named model capabilities, minimal mechanism v0.29.0

17 **Attention backend infrastructure standardization** IMPROVED 45

Adds a `KVCacheLayout` enum to standardize KV cache layout across backends, and a JIT warmup provider registry for attention backends.

— Named internal surfaces, no usage-facing detail v0.29.0

18 **Deterministic decode replay tracing** NEW 40

`trace_decode_token_ids` enables deterministic decode replay.

— Named field with minimal explanation of use v0.29.0

└──▷ **BREAKING ON UPGRADE**

! Ten deprecated model architectures are removed.

! FlexOlmo, Olmo3, and Hunyuan V1/VL are migrated to the Transformers modeling backend, removing their native implementations.

! The PyAV video decoder backend is removed.

! `python -m vllm.entrypoints.openai.api_server` is deprecated in favor of `vllm serve`.

! The `VLLM_TEST_FORCE_FP8_MARLIN` environment variable is removed.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

AWS Context Ontology Accelerator

Feedback ⓘ

1 RELEASE · 2026-09-08 CODE ↗ RANK

Context Ontology Accelerator provides an open-source semantic layer that helps AI agents make consistent, explainable decisions using formal ontologies and knowledge graphs.

> Context Ontology Accelerator v0.3.0 adds a Databricks connector for Athena Federation, letting COA query Databricks Unity Catalog via federated SQL through a new CDK construct.

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

01 **Databricks connector for Athena Federation** NEW 85

A new `AthenaFederationConnector` CDK construct, shipped in the `coa-connector-cdk` package, lets you build and deploy COA connectors as Lambda-backed stacks using helper functions `functionNamePrefix`, `optionalEnv`, `optionalIntEnv`, `queryRoleArns`, and `requiredEnv`. It includes an `InformationSchemaReader` that reads Databricks Unity Catalog's `information_schema` to surface column comments and key constraints that standard JDBC metadata handlers omit, enabling federated SQL queries against Unity Catalog through Athena Federation.

— Names construct, package, and five helper functions but no usage example. v0.3.0

WAS THIS USEFUL? 👍 👎

okf-memory OKF Agent Memory

Feedback ⓘ

1 RELEASE · 2026-09-09 NOTES ↗ RANK

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

> OKF Agent Memory v0.1.5 introduces an autonomous Google Jules adversarial audit review-and-merge workflow, alongside deeper symlink boundary hardening and centralized metadata/ID validation across the CLI and MCP handlers.

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

01 **Jules adversarial audit review loop** NEW 90

Adds `make jules-review` to build, test, and verify pending Jules remediation branches inside temporary, isolated Git worktrees without altering the active working directory; `make jules-merge` to merge verified adversarial audit remediations into `develop` with automated quality and OKF v0.2 conformance gates; and `make jules-list` to discover pending `security-audit-*` and `jules-*` branches across remotes. The adversarial audit model and Jules lifecycle are documented in `convention/security-audit.md`.

Safely review a pending Jules security-audit branch in an isolated worktree before merging it into your main branch.

```
$ make jules-review
```

See all pending adversarial audit branches across remotes before deciding which to review or merge.

```
$ make jules-list
```

Merge a verified Jules remediation into `develop` with OKF v0.2 conformance gates enforced automatically.

```
$ make jules-merge
```

— Names three runnable make targets with clear mechanism and gating. v0.1.5

02 **Symlink boundary containment hardening** IMPROVED 73

Adds ancestor symlink resolution in `resolveBundleDir` (`cmd/okf/mcp.go`) to enforce strict server root containment even when target bundle paths do not yet exist, and symlink target inspection in `resolveInBundle` to verify targets are legitimate `.md` files within the bundle that do not escape to arbitrary system files. Adds case-insensitive reserved document protection for `index.md` (subdirectories), root `log.md`, and root `AGENTS.md` across logical paths and physical symlink targets.

— Internal security fix, mechanism named but no user action. v0.1.5

---

03 **Concept metadata and ID validation hardening** IMPROVED

70

Centralizes `type`, `title`, and `description` validation inside `SaveConcept` via `sanitizeConceptMetadata` in `pkg/okf/mutate.go`, eliminating redundant validation across CLI and MCP handlers. Enforces self-referencing loop prevention in `RelateConcepts` by blocking relationships where `sourceID == targetID` after trimming whitespace and `.md` extensions, adds automatic `actor` fallback to `'agent/okf-tool'` when the `actor` argument is empty or whitespace-only, and trims surrounding whitespace from concept IDs in MCP handlers `okf_show`, `okf_create`, `okf_update`, and `okf_relate` before map lookups.

— Names exact functions and files but no user-facing action. v0.1.5

THINNER COVERAGE BELOW

04 **Coding standards documentation** NEW

32

Documents Clean Code, TDD, DRY, zero-dependency engineering, and Go idioms in `convention/coding-standards.md`.

— Only a file name and topic list, no detail. v0.1.5

WAS THIS USEFUL?  

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ 1 RELEASE · 2026-08-20

NOTES ↗ RANK

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

> LangSmith's release centers on evaluator tooling and annotation workflows, adding CLI LLM-as-judge evaluator creation, a new annotation-queue items endpoint supporting mixed RUN/THREAD batches, and a paginated experiment-comparison API that retires legacy dataset-comparison helpers, alongside bulk dataset-split management in the experiment UI and new permission controls.

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

01 Annotation queue items endpoint for runs and threads NEW

95

Adds `POST /annotation-queues/<id>/items` endpoint to add RUN items to a single annotation queue, with the server resolving runs via ClickHouse or SmithDB and returning a standards-shaped items envelope; the endpoint also supports `item_type` `THREAD` (with `thread_id` + `session_id`) so conversation threads can be queued, with mixed RUN and THREAD batches allowed in one request. Requests are capped at a maximum of 200 items per request, returning a clear validation error when the limit is exceeded.

Add a batch of conversation threads to an annotation queue for human review, mixing RUN and THREAD items in a single request.

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

— Endpoint, resolution mechanism, and item limit all named snapshot-20260909

02 CLI evaluator creation and in-place updates NEW

90

Adds `langsmith evaluator create-llm` CLI subcommand to define structured LLM-as-judge evaluator rules from a prompt, schema, and model config file, targeting a project or dataset. The `evaluator upload --replace` flag on the LangSmith CLI now updates existing code evaluator rules in place, avoiding a delete-before-create window if the replacement upload fails.

— Names exact CLI subcommand and flag with their behavior snapshot-20260909

03 **Thread evaluator testing before save** NEW 90

`POST /runs/rules/validate` now accepts `test_thread_id` and `session_id` to test a thread evaluator against a real conversation before saving. Multi-turn evaluators also gain a Test action in the UI to run the evaluator against a sample thread before committing the rule.

Test a multi-turn thread evaluator against a real conversation before committing the rule, to confirm it grades correctly on live data.

```
$ curl -X POST
'https://api.smith.langchain.com/runs/rules/validate' \
  -H 'Content-Type: application/json' \
  -d '{"test_thread_id": "<thread_id>", "session_id": "
<session_id>", ...}'
```

— Endpoint parameters and matching UI action both named snapshot-20260909

04 **Legacy feedback formula endpoints scheduled for removal**

DEPRECATED 80

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

— Exact endpoints and removal date given snapshot-20260909

05 **Paginated experiment comparison API replaces legacy helpers**

75

BREAKING

`POST /v2/datasets/<id>/experiment-runs` is now the supported public API for paginated experiment comparison; legacy dataset comparison helper endpoints are removed from the public OpenAPI spec and generated SDKs, requiring callers to migrate to the new endpoint.

— New and removed endpoints named, no migration steps given snapshot-20260909

THINNER COVERAGE BELOW

06 **Download datasets permission** NEW 55

Introduces a separate `download datasets` permission, enforced in the application and APIs; the download button is disabled in the UI for users who lack it.

— Names permission but not full enforcement scope snapshot-20260909

07 **Overflow dropdown for config chips in Experiments table**

IMPROVED 55

Model, prompt, and tool chips in the Experiments table config cells now feature a clickable +N overflow badge dropdown exposing filter, group-by, open-in-playground, and details actions.

— UI action named, no deeper mechanism described snapshot-20260909

08 **Image thumbnails and text truncation in tables** IMPROVED

45

Dataset and experiment tables now detect and render base64 images as small thumbnails with a delayed larger preview, and truncate long input and reference-output text, avoiding oversized hidden DOM content.

— Describes behavior, no config or limits named snapshot-20260909

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

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

— No config surface or default numbers specified snapshot-20260909

10 **Longer execution time for code evaluators** IMPROVED

30

Code evaluators now receive more execution time per batch, reducing timeouts for evaluators that import heavy libraries such as scikit-learn.

— No concrete time limit or numbers given snapshot-20260909

└──▷ **BREAKING ON UPGRADE**

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

! Legacy dataset comparison helper endpoints are removed from the public OpenAPI spec and generated SDKs; use `POST /v2/datasets/<id>/experiment-runs` for paginated experiment comparison instead.

WAS THIS USEFUL? 👍 👎

Weights & Biases

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

RANK

Weights & Biases tracks experiments, sweeps, artifacts, and model versions for training runs, and hosts the registry teams promote models through.

> Weights & Biases v0.30.0 adds a non-interactive login check, a new ARIA automation action, and several LEET TUI usability improvements, alongside breaking removals of legacy API classes and CLI sync flags.

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

01 **Non-interactive login credential check** NEW 85

Adds wandb.login(prompt=False) to check whether the user is logged in using configured credentials, returning `False` without prompting for an API key or switching to offline mode — useful for guarding scripts in CI or automated pipelines.

Guard a script block that needs W&B credentials without blocking on an interactive prompt — useful in CI or automated pipelines.

python

```
import wandb

if not wandb.login(prompt=False):
    raise SystemExit("W&B credentials not configured. Set
WANDB_API_KEY or run `wandb login`.")

with wandb.init(project="my-project") as run:
    run.log({"metric": 0.99})
```

— Runnable example shows exact usage and behavior v0.30.0

02 **Console log filtering in LEET TUI** NEW 80

Adds console log filtering to the W&B LEET TUI: with the logs pane focused, press `/` and type a regex pattern, or press `Tab` then type a glob, to show only matching lines; new matches are followed live and `ctrl+/` clears the filter.

— Exact key bindings and behavior given, no code example v0.30.0

03 **wandb sync CLI flag changes** BREAKING 75

The `wandb sync --sync-all` flag has been removed; run `wandb sync` with no arguments instead. The `wandb sync --job_type` flag (underscore) has been removed; use `--job-type` (hyphen) instead.

— Exact flag names and migration path given v0.30.0

04 **Custom x-axis charts in LEET TUI** NEW 70

Adds support for charts with custom x-axes set via run.define_metric(...) in the W&B LEET TUI, including glob definitions such as run.define_metric('train/*', step_metric='train/step'); applies to runs opened from local `.wandb` files.

— Names the API call but no full walkthrough v0.30.0

05 **Removal of legacy Api classes** BREAKING 70

`wandb.InternalApi`, `wandb.PublicApi`, `wandb.apis.PublicApi`, `wandb.apis.internal`, and `wandb.sdk.internal.internal_api` have been removed; replace all usages with wandb.Api().

— Names all removed classes and the replacement v0.30.0

06 **Persisted filters in LEET TUI** NEW 65

Persists metrics, system metrics, and runs filters per wandb directory in `.wandb-leet.json`, restoring them automatically the next time the directory is opened in either the workspace or single-run view.

— Names storage file but no interaction steps v0.30.0

THINNER COVERAGE BELOW

07 **SendPromptToAria automation action** NEW 45

Adds `SendPromptToAria` to the automations API, enabling sending a prompt to ARIA as an automation action.

— Named but no usage example or config shown v0.30.0

└──▷ **BREAKING ON UPGRADE**

! `wandb.InternalApi`, `wandb.PublicApi`, `wandb.apis.PublicApi`, `wandb.apis.internal`, and `wandb.sdk.internal.internal_api` have been removed; replace all usages with wandb.Api().

! The `wandb sync --sync-all` flag has been removed; run `wandb sync` with no arguments instead.

! The `wandb sync --job_type` flag (underscore) has been removed; use `--job-type` (hyphen) instead.

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ 3 RELEASES · 2026-09-08 NOTES ↗ RANK

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

> Arize Phoenix added error/latency filtering to its trace-retrieval APIs and expanded its integration and model-provider surface with AG2, MiniMax, ATIF/Harbor tracing, and Meta's Muse Spark models.

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

01 **Error and latency filters on trace retrieval** NEW 91

Adds `error` and `latency` filter parameters to `get_traces` in the Python and TypeScript clients for targeted trace retrieval. This was extended with `error`, `minLatencyMs`, and `maxLatencyMs` filter parameters on `getTraces` in the JS client, matching the query parameters on `GET /v1/projects/{id}/traces` (requires Phoenix server >= 20.8.0).

— Names exact params, method and path, and version requirement

`@arizeai/phoenix-client@7.10.0` `arize-phoenix-client-v3.5.0`

THINNER COVERAGE BELOW

02 **New integrations and model providers** NEW 55

Phoenix added several new integration and provider surfaces: `AG2` integration for tracing and observability within the platform, `MiniMax` as a supported LLM provider, `ATIF` tracing support added to the Harbor plugin, and `META` as a built-in OpenAI-compatible model provider enabling Meta Muse Spark models to be used directly in Phoenix without manual provider configuration.

— Names each provider/integration but gives no setup steps `@arizeai/phoenix-cli@1.18.0`

`arize-phoenix-v20.9.0` `arize-phoenix-client-v3.5.0`

WAS THIS USEFUL? 👍 👎

PromptLayer

Feedback ⓘ 1 RELEASE · 2026-09-08 NOTES ↗ RANK

PromptLayer is a platform for managing, evaluating, and observing prompts and LLM applications.

> PromptLayer added support for Google Gemini 3.8 Flash across three providers with configurable reasoning effort, and fixed video playback in the Playground's fullscreen view.

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

01 Gemini 3.8 Flash support with reasoning effort levels NEW

75

PromptLayer added support for Google Gemini 3.8 Flash across Google AI Studio, Vertex AI, and OpenRouter, with configurable reasoning effort levels. In the Playground, users can select Gemini 3.8 Flash under any of the three providers and set the reasoning effort level (e.g. high) in the model configuration panel before running a prompt.

Select Gemini 3.8 Flash with high reasoning effort in the Playground to tackle complex agent workflows requiring long-context reasoning.

> 📍 In the Playground, open the model selector and choose Gemini 3.8 Flash under the Google AI Studio, Vertex AI, or OpenRouter provider. Set the reasoning effort level to high in the model configuration panel before running your prompt.

— Names providers and exact selection steps, lacks effort level enumeration

September 08, 2026

THINNER COVERAGE BELOW

02 Video playback in Playground fullscreen view IMPROVED

23

Enhances the Playground fullscreen view to properly display video content and playback controls.

— Bare description of a fix, no mechanism or navigation given September 08, 2026

WAS THIS USEFUL? 👍 👎

Braintrust

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

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

> Braintrust's biggest changes this window are concurrent eval execution with new lifecycle controls in the Java SDK, a public preview of scheduled Loop investigations called Patterns, and a managed-runtime upgrade to Loop, alongside new built-in models and OpenAI Batch tracing support in the TypeScript SDK.

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

01 Concurrent eval execution and concurrency controls in Java SDK

95

BREAKING

Eval cases now run concurrently (default: 3 at a time) instead of sequentially. Adds Eval.start() to launch an eval in the background, returning an `EvalResult` immediately, while Eval.run() still awaits completion; `EvalResult` exposes getStatus(), isDone(), awaitCompletion(), getStartedAt(), getDuration(), getCasesExecuted(), and getAbortCause(). Adds .maxConcurrency(int) on `Eval.Builder` and `Devserver.Builder` and .executor(Executor) to supply a custom thread pool, plus global defaults via environment variables `BRAINTRUST_DEFAULT_MAX_CONCURRENCY`, `BRAINTRUST_OTEL_MAX_QUEUE_SIZE`, `BRAINTRUST_OTEL_MAX_EXPORT_BATCH_SIZE`, and `BRAINTRUST_OTEL_EXPORT_INTERVAL_MILLIS`. Because eval cases now run concurrently, `Task`, `Scorer`, and `Classifier` implementations in the Java SDK (v0.3.23) must now be thread-safe.

Cap eval parallelism and supply a custom thread pool when running evals via the dev server, to control resource usage in CI.

```java
Devserver.Builder builder = Devserver.builder()
    .maxConcurrency(5)
    .executor(myCustomExecutor);
```

Set global concurrency and OTel batch defaults for all evals without code changes, useful for tuning throughput in production pipelines.

```
$ export BRAINTRUST_DEFAULT_MAX_CONCURRENCY=10
export BRAINTRUST_OTEL_MAX_QUEUE_SIZE=2048
export BRAINTRUST_OTEL_MAX_EXPORT_BATCH_SIZE=256
export BRAINTRUST_OTEL_EXPORT_INTERVAL_MILLIS=3000
```

— named env vars, methods, runnable examples, and breaking requirement `snapshot-20260909`

02 **Loop upgraded to managed runtime with persistent threads**

IMPROVED

65

Loop now runs in a Braintrust-managed runtime with persistent threads and cross-project scope (logs, experiments, datasets), and can create and edit prompts, scorers, datasets, facets, custom views, dashboards, and automations with per-write approval or auto-accept.

— mechanism described but no exact API or command given `snapshot-20260909`

03 **Patterns: scheduled Loop investigations (public preview)** NEW

60

Adds Patterns, scheduled Loop investigations that surface recurring failure modes, cost trends, and cohort-level problems across project traces, saving each finding with supporting traces and suggested fixes.

— describes mechanism but no config or command shown `snapshot-20260909`

THINNER COVERAGE BELOW

04 **org_name parameter for cross-org name resolution** NEW

55

Adds `org_name` parameter to pattern, prompt, and scorer resolution tools, and to Loop and the MCP server, to disambiguate name lookups across organizations — resolving members, API keys, service tokens, patterns, prompts, and scorers by name instead of raw IDs.

— names the parameter and affected surfaces but no call example `snapshot-20260909`

05 **Shareable fullscreen links for dashboard charts** NEW

45

Adds a shareable fullscreen link for individual dashboard charts; chart move and copy actions now appear directly in each chart's more-options menu.

— UI feature with clear navigation but no API `snapshot-20260909`

06 **Timezone selection for automation cron schedules** NEW

40

Adds timezone selection for custom cron schedules in automations; new Pattern discovery automations default to running daily at 9:00 AM.

— UI setting described, no config key named `snapshot-20260909`

07 **GPT-6 Astra model in Loop's model picker** NEW 35

Adds GPT-6 Astra (multimodal reasoning) to Loop's model picker, including scheduled automations.

— names the model but gives no usage detail `snapshot-20260909`

08 **Faster large trace downloads with progress indicator** IMPROVED 35

Large traces now download faster from the trace viewer, with a progress indicator showing span count fetched and percentage complete.

— describes UI behavior with no mechanism or numbers `product docs`

09 **Faster loading for long Loop threads** IMPROVED 35

Long Loop threads now load faster by surfacing recent history first, with earlier messages available on demand.

— describes behavior with no metrics or config `product docs`

└──▷ ALSO FROM THESE RELEASES

Filter a CI PR comment to show only specific scores and metrics — useful when your eval suite produces many signals but you only want pass-rate and latency surfaced in the pull request.

```yaml
- uses: braintrustdata/eval-action@v2.1.0
  with:
    api_key: ${{ secrets.BRAINTRUST_API_KEY }}
    report_scores: accuracy, pass_rate
    report_metrics: latency, cost
```

└──▷ BREAKING ON UPGRADE

! `Task` and `Classifier` implementations must now be thread-safe due to concurrent eval case execution.

! `Task`, `Scorer`, and `Classifier` implementations in the Java SDK (v0.3.23) must now be thread-safe due to concurrent eval case execution.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

NoLabs nono

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

RANK

The nono runtime runs AI agents in isolated environments with policy-controlled filesystem, network, and credential access.

> nono's latest releases focus on operational hygiene for its CLI and sandboxing model: a new audit cleanup subcommand for managing on-disk session storage, scoped Unix socket grants for sandboxed commands, and Nix packaging support.

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

01 Audit session cleanup subcommand NEW

90

nono audit gains a `cleanup` subcommand that removes old audit sessions from disk, with filters that can be combined to narrow results: `--keep` retains the N newest sessions, `--older-than` removes sessions exceeding a day threshold, `--max-total-size` evicts oldest-first until total usage is under a given number of megabytes, and `--dry-run` previews removals without deleting anything.

Preview which audit sessions would be removed to keep the 10 newest and stay under 500 MB, without deleting anything.

```
$ nono audit cleanup --keep 10 --max-total-size 500 --dry-run
```

Enforce a retention policy in CI by purging sessions older than 30 days and capping total storage at 500 MB.

```
$ nono audit cleanup --older-than 30 --max-total-size 500
```

Cap total audit storage at 500 MB, automatically pruning the oldest sessions to stay under the limit — useful in CI environments where audit logs accumulate unbounded.

```
$ nono audit cleanup --max-total-size 500
```

— Full flag set and runnable commands given, mechanism explained. v0.76.0

02 Unix socket grants in tool sandboxes NEW

80

Adds `unix_socket_bind` field to `CommandSandboxConfig` in tool-sandbox profiles, enabling scoped Unix domain socket grants (connect or bind) for individual sandboxed commands such as Git fsmonitor, without widening the rest of the sandbox.

Grant a sandboxed `git` command access to its fsmonitor daemon socket without widening the rest of the sandbox — place this in your tool-sandbox command policy.

json

```
{
  "command_policies": {
    "commands": {
      "git": {
        "from": {
          "session": {
            "sandbox": {
              "fs_read": ["."],
              "unix_socket_bind": ["/tmp/git-fsmonitor-
<pid>.sock"]
            }
          }
        }
      }
    }
  }
}
```

— Named config field with example policy snippet, but scope of use is narrow. v0.76.0

THINNER COVERAGE BELOW

03 **Nix flake installation support** NEW 30

Adds Nix flake support for installing nono via the Nix package manager.

— Bare mention of new install method, no further detail. v0.76.0

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 9, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue