THE AI TOOLCHAIN

NO. 029

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 17, 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 261 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 43 tools

▾ // THE BRIEF HIDE

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

Three unrelated things stand out. Inspect AI can now route a tool's result to a human reviewer before the agent sees it, and fail a generation fusal. ai-gateway gained callable PII and prompt-injection

vim ? help

detectors plus an observe-only mode, so you can measure false positives before blocking. Braintrust's Debugger reads a single failing agent trace and names the failure mode.

GOVERN

Gate what a tool hands back to the agent — with a human in the loop when it matters

Inspect AI 0.3.264 adds configurable tool-result review policies including a human reviewer, plus a `fail_on_refusal` generate option so a refused completion surfaces as a failure instead of scoring as a passing sample. ai-gateway exposes PII and prompt-injection detection as callable functions with an observe-only PII logging mode, letting you baseline hit rates on real traffic before you turn blocking on; nono adds backend-signed approval webhooks so approvals can't be forged client-side.

Inspect AI · ai-gateway · nono

EVALUATE

Find out why one agent run failed without reading the whole trace by hand

Braintrust's Debugger diagnoses failure modes in an individual agent trace, and Loop now runs on a managed runtime for longer, cross-surface investigations — the alternative is scrolling spans looking for the turn where the agent went wrong. MLflow 3.16.1 adds scorer execution timeouts and records scorer versions in evaluation metadata, so a hung LLM judge no longer stalls a whole eval run and score shifts can be traced to a changed scorer rather than a changed model.

Braintrust · MLflow

DEPLOY

Get more tokens per second out of hardware you already own

OpenVINO 2026.4 adds two new speculative-decoding techniques for GenAI — a small draft model proposes tokens the large one verifies in one pass, same output, less wall clock — plus full ITT/VTune profiling on NPUs, which is the first honest way to see where NPU time actually goes. ExecuTorch v1.5.0 ships a C++ SDK with linkable delegate libraries, CUDA wheels and new quantized LLM/MoE operators; mlx-serve adds a greedy-decoding shortcut and an in-app benchmark.

OpenVINO

BUILD

◆ AI Coding Agents

Cline

Feedback ⓘ 2 RELEASES · 2026-09-16 → 2026-09-17 NOTES ↗ CODE ↗

RANK

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

> Cline's latest updates focus on safer model handling and finer-grained control during agent sessions, alongside a major shift in default model resolution across dozens of providers.

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

01 **Steer queued messages mid-turn with Enter** NEW 75

Pressing Enter with an empty composer while the agent is running steers the first queued message into the active turn. The steer is claimed atomically so a fast second Enter cannot steer a message that has already left the queue, and failed steers surface as a toast instead of silently doing nothing.

– Explains exact trigger and concurrency safeguard, no command name desktop-v0.0.29

THINNER COVERAGE BELOW

02 **Warning on image attachments to text-only models** IMPROVED 55

Cline now flags images attached to a text-only model with a warning badge and offers a one-click switch to an image-capable model, rather than silently discarding the attachment.

– Describes behavior change and UI fix but no exact model list v4.1.19

03 **Session sidebar sort toggle icon fix** IMPROVED 35

The session sidebar sort toggle now shows the mode a click will switch to (folder icon when sorted by time, clock icon when grouped by project) rather than the mode already active.

– Thin UI fix, no broader mechanism described desktop-v0.0.29

└──▷ BREAKING ON UPGRADE

! The resolved default model changes for 44 providers — most landing on DeepSeek V4.1 Flash; Gemini and Vertex now resolve to Gemini 3.8 Flash, GitHub Copilot and Vivgrid to GPT-6 Astra, and NVIDIA to GLM 5.3 Flash. Any provider used without a pinned model will switch to a different default after this update.

WAS THIS USEFUL?  

All Hands AI OpenHands

Feedback ⓘ

2 RELEASES · 2026-09-16 → 2026-09-17 NOTES ↗ RANK

OpenHands is an open-source software-development agent that writes code, runs commands, and browses the web.

> OpenHands shipped finer-grained control over agent profiles — scoping secrets and enabling saved-profile selection in automations — alongside Docker runtime forwarding, GPT-6 Astra model support, and new UI visibility into automation disablement.

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

01 **Per-profile secret scoping** NEW 45

Agent profiles can now be configured with a specific set of secrets available to them, scoping credential exposure so each profile only has access to the secrets assigned to it.

— Describes mechanism but no config key or UI path given v1.20.0

02 **Docker runtime settings forwarding** IMPROVED 45

Docker conversation runtime settings are now forwarded to the agent runtime, propagating runtime configuration from the conversation session into the agent's runtime environment.

— Mechanism named but no specific settings enumerated v1.20.0

03 **Saved agent profiles in automations** NEW 35

Automations can now be configured to use a saved agent profile, letting users assign a specific profile when setting up an automation.

— Capability named but no exact setup steps v1.20.0

04 **GPT-6 Astra model support** NEW 35

OpenHands now supports the GPT-6 Astra model as an available agent model.

— Names the model but no setup or config detail v1.19.0

05 **Automation disablement reason in UI** IMPROVED 35

The UI now surfaces the reason an automation was disabled, giving users visibility into why an automation stopped running.

— Explains benefit but not exact UI location v1.19.0

06 **Authenticated Canvas arrivals tracking** NEW 10

OpenHands now tracks authenticated Canvas arrivals.

— Bare mention with no context or mechanism v1.19.0

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ 4 RELEASES · 2026-09-16 → 2026-09-17

NOTES ↗ CODE ↗ RANK

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

> OpenAI Codex CLI shipped a hidden `tcp-tunnel` subcommand for TLS-verified HTTP/3 proxying, a `--no-daemon` flag to bypass the shared background server, and a `/daemon` TUI menu for updating that server, alongside smaller TUI rendering, MCP access-control, and macOS configuration-security changes.

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

01 `/daemon` TUI menu for background server updates NEW

95

Adds a `/daemon` menu in the TUI to update the local background app-server daemon, offering a choice between the latest public stable release (`DaemonUpdateSource::PublicStable`) or the current CLI package (`DaemonUpdateSource::ThisCli`), with confirmation and failure propagation.

Trigger a daemon update from inside the TUI without dropping to a shell — useful when Codex warns that the local background server is out of date.

```
$ /daemon
```

— Exact menu command and update-source options with runnable example

rust-v0.155.0-alpha.11

02 `--no-daemon` flag to bypass background server NEW

85

Adds `--no-daemon` flag to run Codex CLI without the shared background server, even if one is already running; it is incompatible with `codex agents` and `--remote`.

Start an isolated Codex session that skips the shared background server — useful in ephemeral CI environments or when you want a fully self-contained process.

```
$ codex --no-daemon
```

— Named flag, incompatibility, and runnable command given rust-v0.155.0-alpha.16

03 Hidden `tcp-tunnel` subcommand for TLS-verified proxying NEW

85

Adds a hidden `codex tcp-tunnel` subcommand (internally `tcp-tunnel`, also implemented as `TcpTunnel`) to forward loopback/local TCP connections to an explicit target through a TLS-verified HTTP/3 CONNECT proxy, with bearer-token and optional `x-` header credentials read from stdin, token-update support, and `LISTENING` / `AUTH_UPDATED` notifications.

— Names subcommand, auth mechanism, notifications; still hidden/no example

`rust-v0.155.0-alpha.15` `rust-v0.155.0-alpha.11`

---

04 **Enforced macOS managed-configuration validation** IMPROVED

75

Requires `CFPreferencesAppValueIsForced` checks before reading `config_toml_base64` and `requirements_toml_base64` macOS managed-configuration preference keys, preventing user-writable domains from supplying trusted administrator configuration.

— Names exact keys and check function, no user-facing command `rust-v0.155.0-alpha.14`

---

05 **`?` wildcard support in proxy domain patterns** IMPROVED

70

Documents and tests `?` wildcard support in network proxy allow/deny domain patterns, where `?` matches exactly one character (including a dot) — e.g. `api?.example.com` matches `api1.example.com` but not `api.example.com`.

— Concrete pattern syntax and matching example given `rust-v0.155.0-alpha.14` `rust-v0.155.0-alpha.15`

THINNER COVERAGE BELOW

06 **Tool exposure and read-only policy controls for MCP** NEW

35

Adds per-app tool exposure configuration and read-only policy support for MCP tool requests.

— Names the controls but no keys or commands given `rust-v0.155.0-alpha.15`

---

07 **Mermaid diagram rendering in TUI** NEW

30

Renders Mermaid code blocks as diagrams in the TUI.

— Brief description, no mechanism or example `rust-v0.155.0-alpha.15`

08 **TUI animations disabled for screen readers** IMPROVED

30

Defaults TUI animations off when a screen reader is detected.

— States behavior but no configuration detail `rust-v0.155.0-alpha.15`

09 **Configurable F8 shortcut for voice conversations** NEW

25

Adds a configurable F8 shortcut for voice conversations.

— Only mentioned in summary, no config detail `rust-v0.155.0-alpha.16`

10 **Summary history fallback for read-only conversations** IMPROVED

20

Falls back to summary history for read-only conversations.

— One-line change with no further detail `rust-v0.155.0-alpha.15`

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ 1 RELEASE · 2026-09-17 NOTES ↗

RANK

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

> Qwen Code adds a Linux `bwrap` sandbox backend for OS-level tool isolation, a `+Nk` per-turn token budget directive, hook registry hot-reloading, and several web shell UI refinements.

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

01 **Per-turn token budget directive** NEW 80

Prepend a `+Nk` directive to a prompt to cap how many tokens a workflow can consume per turn, preventing runaway multi-step runs from exhausting your budget, e.g. `qwen -p "+50k Run the full /batch refactor workflow across all packages"`.

Cap how many tokens a workflow can consume per turn by prepending a +Nk directive, preventing runaway multi-step runs from exhausting your budget.

```
$ qwen -p "+50k Run the full /batch refactor workflow across all packages"
```

— Includes an exact runnable command demonstrating the syntax. desktop-v0.24.0

02 **Hook registry reload on `/hooks` open** IMPROVED 65

Reloads the hook registry when `/hooks` opens, picking up newly added hooks without restarting.

— Names the exact command but no further config detail. desktop-v0.24.0

THINNER COVERAGE BELOW

03 **bwrap kernel sandbox backend for Linux** NEW 50

Adds a `bwrap` (bubblewrap) sandbox backend for Linux, providing OS-level process isolation for tool execution.

— Names the backend but no flag or config to enable it. desktop-v0.24.0

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ 1 RELEASE · 2026-09-17

NOTES ↗ CODE ↗ RANK

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

> Claude Code's v2.1.274 release focuses on infrastructure and observability, tightening MCP startup and connection handling, expanding OpenTelemetry and gateway telemetry, and adding several small VSCode extension conveniences, alongside a default shift to the v2 MCP client with new protocol negotiation.

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

01 MCP startup wait cap for non-interactive turns NEW

85

Adds `CLAUDE_CODE_MCP_STARTUP_WAIT_MS` environment variable to cap how long the first non-interactive turn waits for MCP servers to connect, with `0` meaning don't wait at all.

Prevent CI non-interactive sessions from stalling at startup waiting for slow MCP servers to connect.

```
$ CLAUDE_CODE_MCP_STARTUP_WAIT_MS=0 claude -p 'run the test suite and summarise failures'
```

— Exact env var and effect on wait behaviour given with runnable example. v2.1.274

02 Configurable Postgres connect timeout in Claude apps gateway

NEW

85

Adds `store.connect_timeout_seconds` to the Claude apps gateway config to extend the Postgres connect timeout beyond its 5-second default; the boot error now points to `store.postgres_url` and the configured timeout.

Lengthen the Postgres connect timeout on a Claude apps gateway where the database takes longer than 5 seconds to become reachable at boot.

yaml

```
store:
  connect_timeout_seconds: 30
  postgres_url: postgres://user:pass@db-host:5432/claude
```

— Config key, default value, and boot error behaviour all named with example. v2.1.274

03 Expanded OpenTelemetry and gateway observability IMPROVED

85

Adds a `claude_code.managed_settings_resolved` OTel event exposing managed-settings sources and policy helper state, viewable with redacted values and digests via `OTEL_LOG_MANAGED_SETTINGS=1`. Adds an `effort` attribute to the `claude_code.llm_request` trace span (matching the existing `api_request` event) and `enduser.sub` (the IdP subject) to telemetry sent by Claude Desktop and Cowork through a Claude apps gateway. The gateway now warns and logs a startup line when a replica has more than 256 requests open upstream at once. `OTEL_LOG_RAW_API_BODIES=file:<dir>` output gains an `index.jsonl` file plus `request_body_id` / `message.id` attributes linking each response to its request file and transcript message.

Expose redacted managed-settings values and digests in OTel logs to audit which policy sources are active in a managed deployment.

```
$ OTEL_LOG_MANAGED_SETTINGS=1 claude
```

— Names every event, attribute, env var and threshold but most are diagnostic-only.

v2.1.274

04 v2 MCP client and new protocol negotiation by default

80

BREAKING

Bedrock, Vertex, Foundry, and telemetry-disabled installs now use the v2 MCP client and MCP 2026-07-28 negotiation with direct HTTP servers by default; opt out with `MCP_SDK_GENERATION=v1` or `MCP_PROTOCOL_NEGOTIATION=legacy`.

— Names affected installs, protocol version, and both opt-out env vars. v2.1.274

05 VSCode extension additions: editor locking, memory UI, reload continuation

NEW

75

Adds `claudeCode.lockEditorGroups` setting to prevent Claude from locking the editor groups it opens in. Adds Memory and Instructions entries to the Customize menu, where Memory shows auto-memory toggles, saved memories, and memory folders, and Instructions edits CLAUDE.md files. Adds continuation of a step interrupted by a window reload, labeled in the chat, controlled by a `Claude Code: Continue After Reload` setting.

— Names three settings and menu locations but only a UI path, not commands. v2.1.274

06 SDK-type MCP entries no longer register from files

BREAKING

60

`"type": "sdk"` MCP entries in `.mcp.json`, settings, plugins, and agent files are now skipped with a warning; only an SDK host application can register in-process servers.

— Names affected files and behaviour but no migration path given. v2.1.274

THINNER COVERAGE BELOW

07 **Leaner inline prompts for /code-review** IMPROVED

55

Changes `/code-review` to use leaner inline review prompts for every model without tuned settings, instead of spawning many review subagents.

— Names the command and the before/after behaviour, no further detail. v2.1.274

08 **Click-to-expand collapsed messages in fullscreen** IMPROVED

35

Adds click-to-expand for collapsed teammate and agent messages in fullscreen mode.

— Bare description, no mechanism or navigation detail. v2.1.274

└──▷ **BREAKING ON UPGRADE**

! Bedrock, Vertex, Foundry, and telemetry-disabled installs now use the v2 MCP client and MCP 2026-07-28 negotiation with direct HTTP servers by default; set `MCP_SDK_GENERATION=v1` or `MCP_PROTOCOL_NEGOTIATION=legacy` to opt out.

! `"type": "sdk"` MCP entries in `.mcp.json`, settings, plugins, and agent files are now skipped with a warning — only an SDK host application can register in-process servers.

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

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

> Letta Code adds a new `letta usage` command so developers can check credits and model quota directly from the CLI.

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

01 **`letta usage` command for credits and quota** NEW 65

Adds a `letta usage` command that surfaces remaining credits and model quota directly from the CLI, letting developers check usage before running long agent tasks.

Check remaining credits and model quota before kicking off a long-running agent task.

```
$ letta usage
```

— Names exact command but no detail on quota mechanics v0.32.12

WAS THIS USEFUL? 👍 👎

herdr

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

Herdr is a coding-agent runtime that manages persistent terminal sessions across local and SSH machines.

> Herdr's v0.9.1 window centers on a `--machine` flag that routes agent, pane, workspace, and worktree commands to saved SSH machines without opening a TUI, extends remote attach and SSH connectivity to Windows hosts, and ships an experimental Letta Code integration with native conversation resume, alongside sidebar navigation, token-hiding, and session-recovery improvements.

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

01 Remote CLI command routing via --machine flag NEW

95

The `--machine <label-or-id>` global prefix routes supported CLI commands — `api`, `snapshot`, `status`, `server`, and API-backed plugin commands, as well as agent, pane, workspace, and worktree control — to a saved remote SSH machine over the JSON API without opening a TUI, using CLI forwarding; failed remote commands never fall back to Local. Local installation/configuration commands, plugin installation, session management, and interactive terminal attachment are not forwarded. Remote worktree paths must be absolute, `~`, or start with `~/` (expanded on the server), plugin link paths must be absolute, and combining `--machine` with `--session` is an error since the saved remote session is used automatically.

Run an API command against a remote machine by its saved label without opening a TUI on that host.

```
$ herdr --machine prod-server api snapshot
```

Check server status on a remote saved machine identified by its profile ID.

```
$ herdr --machine w1:p1 status server
```

Run a remote agent command against a specific machine from a script or CI pipeline without opening the TUI.

```
$ herdr --machine <label-or-id> agent list
```

— Documents exact subcommands, restrictions, and runnable examples v0.9.1

02 **Experimental Letta Code integration** NEW 90

Adds an experimental Letta Code integration installed via CLI only (not the Settings screen or JSON integration API) with `herdr integration install letta` or `herdr agent start --kind letta`; it reports as letta (experimental). Named Letta Code conversations resume natively via `letta --conversation <id>`, with the default conversation recorded as `default:<agent-id>` and resumed via `letta --conversation default --agent <agent-id>`.

Start tracking a Letta Code agent session when the experimental integration is not yet installed, using kind-based detection directly.

```
$ herdr agent start --kind letta
```

Resume a specific named Letta Code conversation after a herdr server restart, using the native session reference herdr recorded.

```
$ letta --conversation default --agent <agent-id>
```

Install the Letta Code integration to enable native conversation restore on a herdr server.

```
$ herdr integration install letta
```

— Exact install and resume commands with conversation naming scheme v0.9.1

03 **Sidebar navigation and workspace switching across machines** NEW 75

Adds a `prefix+w` keyboard shortcut to navigate workspaces across connected machines in sidebar order, with Enter to activate, Esc to cancel, automatic skipping of disconnected machines, and a visible highlight before Enter switches machines. Adds a collapse/expand toggle (click the arrow beside any machine) to show or hide a machine's workspace list in the sidebar without leaving the current workspace.

— Names exact shortcut and UI path for navigation v0.9.1

04 **hide=true token rule for sidebar** NEW 75

Setting `hide = true` on a token rule removes the matching token and its separator from display in the sidebar; rows with no remaining tokens also disappear. For example,

`{ token = "machine", fg = "#61afef", rules = [{ equals = "Local", hide = true }] }` suppresses a `Local` machine label from the status bar.

Hide a token entirely when its label matches a specific value, such as suppressing a `Local` machine label from the status bar.

```toml
{ token = "machine", fg = "#61afef", rules = [{ equals = "Local", hide = true }] }
```

Hide a sidebar token category and its separators when no matching tokens remain visible.

```toml
hide = true
```

— Exact config key with runnable config snippet v0.9.1

05 **Windows host support for remote sessions** NEW 70

Remote attach, saved-machine setup, and background saved reconnect now work on Windows hosts, using the installed Windows OpenSSH client and SSH configuration; interactive direct attach and saved-machine setup can install or update the complete Herdr package on Windows hosts, though background saved reconnect only discovers already-installed packages and cannot prompt. Multi-machine connections are supported across Linux, macOS, and Windows clients connecting to Linux/macOS servers on x86_64 or aarch64, or Windows servers on x86_64, with native SSH connectivity to Windows hosts from any of these platforms.

— Names platforms and SSH mechanism but no exact CLI command v0.9.1

06 **Inline text editing in Herdr inputs** NEW 70

Adds full inline text editing to Herdr inputs: character and word movement, Home/End, deletion, and `Ctrl+A` / `E` / `K` / `U` / `W` / `Y` shortcuts, including Unicode text.

— Names exact keyboard shortcuts but no config surface v0.9.1

07 **Session file recovery backups** NEW 70

Corrupted or unreadable session files are preserved as recovery copies in `session-backups/` before being overwritten, logging the recovery path as `persist.backup`. The three most recent recovery copies are retained, with older ones pruned only after a new copy is safely written.

— Names backup directory and log key but no trigger command product docs

08 **Claude integration v10 source matcher scoped** BREAKING

70

The Claude integration v10 source matcher is narrowed from a wildcard to only the documented values `startup`, `resume`, `clear`, `compact`, and `fork`, preventing Grok's `load` events from triggering the Herdr Claude hook. Reinstall the Claude integration after upgrading to migrate off the previous wildcard matcher.

— Names exact matcher values and required migration step product docs

THINNER COVERAGE BELOW

09 **Bridge idle cleanup for saved-machine connections** NEW

55

Adds bridge idle cleanup for saved-machine connections to Linux and macOS: closes the remote bridge after one minute without traffic, with sleep counting toward the deadline.

— Names exact timeout but no config to adjust it product docs

10 **OpenCode V2 support** NEW

55

Adds OpenCode V2 support (tested with beta), installing a V2 TUI entrypoint at `herdr-opencode/tui.js` and registering the V2 TUI plugin in `cli.json` alongside the existing V1 plugin in `tui.jsonc`.

— Names files and configs but no usage command shown product docs

11 **Observer disconnect timeout for stalled sockets** IMPROVED

50

On Linux and macOS, an observer is disconnected if a socket write makes no progress for 30 seconds, and a stalled stream can end without a final record while the pane and other clients keep running.

— Names exact timeout but no way for reader to act product docs

12 **Pane image rendering with graphics disable option** NEW

45

Pane images render by default in compatible outer terminals, with popups, menus, and notifications hiding only overlapping image placements temporarily. Graphics rendering and the pane graphics API can be disabled via configuration.

— Describes behavior but no exact config key named product docs

13 **Ctrl-hover link highlighting** NEW

45

Adds Ctrl-hover link highlighting before opening, with wrapped URLs and OSC 8 links remaining clickable even when their beginning or end is outside the viewport.

— Describes behavior but no command or config to adjust v0.9.1

14 **Batched handoff for 64+ panes** IMPROVED

30

Updated Unix servers can now transfer more than 64 panes in a single handoff by sending them in batches.

— States the change with no mechanism or config detail `product docs`

└──▷ BREAKING ON UPGRADE

! The Claude integration v10 source matcher is narrowed from a wildcard to only `startup`, `resume`, `clear`, `compact`, and `fork`; reinstall the Claude integration after upgrading to migrate the previous wildcard matcher.

WAS THIS USEFUL? 👍 👎

Zed

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

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

> Zed v1.20.1 adds configurable window titles, a Markdown preview mode, cursor movement animation, and Emmet wrap, alongside improvements to command palette history, project panel tooltips, and syntax highlighting.

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

01 **Configurable window title format** NEW 90

Adds `window_title_format` and `window_title_separator` settings for configurable window title formatting, supporting variables `${projectName}`, `${fileName}`, `${filePath}`, `${relativePath}`, `${branch}`, `${remoteName}`, `${remoteHost}`, `${appName}`, and `${separator}`. Imports existing values from VS Code's `window.title` and `window.titleSeparator` settings.

Show branch, remote host, and file path in the window title — useful when working across multiple remote sessions at once.

```json
{
  "window_title_format": "${branch} @ ${remoteHost} — ${filePath}",
  "window_title_separator": " | "
}
```

— Exact config keys, full variable list, and runnable example given. v1.20.1

02 **Command palette history controls and display** IMPROVED 75

Adds `command_palette.use_command_history` setting to disable history-based command palette ranking without erasing history, and improves recent commands display with history indicators, a wider layout, and controls to remove individual commands from history using `shift-backspace` or a remove button.

— Names setting and shortcut but lacks a runnable example. v1.20.1

03 **Filesystem watching debug command** NEW 75

Adds `dev: Debug Filesystem Watching` command to inspect local watcher events, watch roots, and scan exclusions, and to export diagnostic captures as JSON.

— Command named with clear scope, but no usage example. v1.20.1

04 **Markdown preview mode for files** NEW 65

Adds `markdown_preview.open_markdown_files_in_preview` setting to open Markdown files directly in the rendered preview instead of the raw editor.

Open Markdown files straight into rendered preview so docs are always readable without a manual toggle.

```json
{
  "markdown_preview": {
    "open_markdown_files_in_preview": true
  }
}
```

— Names the exact setting with a working config example. v1.20.1

05 **New window default view setting** NEW 65

Adds `on_new_window` setting to choose whether new windows show the Launchpad (`launchpad` , the default) or an empty untitled buffer (`empty_tab`).

— Names setting and both values, no worked example. v1.20.1

06 **Emmet wrap with abbreviation** NEW 60

Adds Emmet wrap with abbreviation support via the `editor: wrap with abbreviation` command.

— Named command, but no example of use shown. v1.20.1

THINNER COVERAGE BELOW

07 **Optional cursor movement animation** NEW 55

Adds optional cursor movement animation via `cursor_animation.enabled` set to `true` .

Enable the cursor movement animation for a smoother editing experience.

```json
{
  "cursor_animation": {
    "enabled": true
  }
}
```

— Named setting with config example, but no mechanism detail. v1.20.1

08 **File path tooltip in Project Panel** NEW 55

Adds a tooltip showing the file path when hovering over files and folders in the Project Panel, configurable via the new `project_panel.title_tooltip_delay` setting to disable or adjust the delay.

— Setting named but no example or default value given. v1.20.1

09 **Syntax highlighting for C23 and Go** IMPROVED 45

Adds syntax highlighting for the C23 `constexpr` keyword and improves Go syntax highlighting by distinguishing control-flow keywords from other keywords.

— Names both languages and the keyword but no examples. v1.20.1

10 **Dynamic document selector routing for language servers** IMPROVED 40

Improves language server request routing to respect dynamically registered document selectors.

— States the mechanism but no way for readers to act. v1.20.1

11 **Git Graph commit timestamp consistency** IMPROVED 40

Improves Git Graph commit details to show the same full commit timestamp as the `Date` column.

— Names the UI element but no navigation path given. v1.20.1

12 **Live repository clone progress** IMPROVED 35

Improves repository cloning by showing live progress in the activity indicator.

— Describes the change but not the underlying mechanism. v1.20.1

13 **Gemini 3.8 Flash model added** NEW 30

Adds Gemini 3.8 Flash to the Google AI models.

— Names the model but no setup or usage detail. v1.20.1

14 **Multi-stroke keybinding countdown indicator** NEW 20

Adds a status bar countdown for pending multi-stroke key bindings.

— Bare description with no configuration or mechanism. v1.20.1

15 **Multi-cursor editing performance** IMPROVED 20

Improves multi-cursor editing performance.

— No numbers or mechanism given, purely a claim. v1.20.1

WAS THIS USEFUL?  

DeepSeek Harness

Feedback ⓘ 1 RELEASE · 2026-09-17 CODE ↗

RANK

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

> DeepSeek Harness alpha.2 consolidates Subagent settings into a single page, adds inline per-plugin configuration, and introduces known-site link icons alongside smaller developer-facing changes.

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

01 **Per-plugin configuration on Plugins page** NEW 55

Plugin configuration is now hosted directly on the Plugins page, with per-row config accessible by clicking the plugin's name or label instead of navigating elsewhere.

Access per-plugin configuration inline — navigate to the Plugins page and click a plugin's name or label to open its configuration row directly.

📍 In the web UI, go to Plugins › find your plugin row › click the plugin name or label to open its configuration panel.

— Exact UI navigation given via usage example dsh-v0.1.6-alpha.2

02 **F12 DevTools keybind for Feishu login** NEW 55

Adds an `F12` keybind on Desktop to open Feishu login DevTools.

— Exact key and target named, directly usable dsh-v0.1.6-alpha.2

03 **Unified Subagent settings UI** IMPROVED 40

Subagent configuration is consolidated into a single unified settings page on the web interface, replacing scattered settings.

— Names the UI area but no exact steps dsh-v0.1.6-alpha.2

04 **Known-site link marks with brand icons** NEW 40

Link displays for mainstream/known sites are now led by the site's own icon, sourced from simple-icons.

— Visual change with named icon source, no user action dsh-v0.1.6-alpha.2

05 **Independent office-to-pdf module** IMPROVED 40

Shared Office conversion logic is decoupled from preview into an independent `office-to-pdf` module.

— Named module but internal refactor, little user action dsh-v0.1.6-alpha.2

06 Ownership lifecycle primitives in client session layer NEW

10

Adds ownership lifecycle primitives to the client session layer.

— Vague description with no mechanism or scope `dsh-v0.1.6-alpha.2`

WAS THIS USEFUL? 👍 👎

Claudoscope

Feedback ⓘ 1 RELEASE · 2026-09-17 NOTES ↗ RANK

Claudoscope is a native macOS menu-bar app that tracks Claude Code sessions, estimates costs, scans secrets, and surfaces analytics.

> Claudoscope v1.2.0 adds cost attribution and health monitoring for Claude Code's skill/MCP/agent ecosystem, plus nine new hardening and config checks covering managed MCP servers, Bash permission wildcards, skill scope collisions, and plugin drift.

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

01 **New config validation checks CFG019, CFG020, CFG021** NEW

80

CFG019 flags a `managedMcpServers` entry that defines a `stdio` command, which Claude Code silently skips — meaning an administrator can believe a server is deployed fleet-wide while it runs nowhere. CFG020 flags a `permissions.allow` Bash rule whose wildcard is not its final segment, a privilege escalation because Bash rules match a prefix and ignore everything after the `*`; checked across user, project, and project-local settings. CFG021 flags `bashOutputMaxChars` or `taskOutputMaxChars` set outside the range Claude Code accepts, which it clamps without reporting so the configured number is not the one in effect.

— Each check's exact mechanism and config keys are named. v1.2.0

02 **New skill validation checks SKL010, SKL011, SKL015, SKL016**

NEW

80

SKL010 flags a relative link in `SKILL.md` that resolves to nothing so instructions Claude was meant to load are silently absent. SKL011 flags a tool restriction naming an MCP server that is not configured, where an allow entry grants no access and a deny entry blocks nothing. SKL015 flags the same skill name defined in two scopes, which Claude Code resolves to one without reporting which, so an edit can land in the copy that is not running. SKL016 flags an installed skill that never appears in an attributed turn while others do, since its description occupies context on every request.

— Each check's failure mode and file involved are named. v1.2.0

03 **Plugin scope drift detection** NEW

75

New check PLG004 flags plugin scope drift — the same plugin installed at user and project scope at different commits. The Plugins rail now badges drifted plugins, lists each scope with its version and commit, and explains the impact on specific projects.

— Names the check and the UI surface but not exact navigation path. v1.2.0

04 **Managed MCP servers as a read source** NEW 70

Claudoscope now reads `managedMcpServers` so managed MCP servers appear in the MCPs rail with a `managed` badge, taking precedence on name collisions to match Claude Code behavior.

— Names the field and badge but not exact UI navigation. v1.2.0

05 **Hardening check for unrestricted file reads (HRD014)** NEW 70

HRD014 flags `permissions.blockReadsOutsideWorkingDirectories` not enabled, which leaves Read, Grep, and Glob able to reach any path on disk including other repositories and credential files.

— Names the exact setting and the risk it prevents. v1.2.0

06 **Containment Escape added to hardening baseline** NEW 70

The hardening auto-mode baseline now covers Containment Escape — cloud metadata-credential fetches, egress-allowlist evasion, and cross-tenant reach — and installing the baseline also satisfies HRD012.

— Names the baseline scope and the check it now satisfies. v1.2.0

07 **Prompt cache health card** NEW 65

A new card under Cache Effectiveness reports tokens re-primed and their cost, a cold-turn floor, and two evidenced prefix-rebuild causes: sessions resumed on a later day and mid-session model switches.

— Explains what the card reports and the causes it evidences. v1.2.0

08 **New settings.json fields surfaced** NEW 60

Claudoscope now reads and displays `maxEffortLevel`, `bashOutputMaxChars`, `taskOutputMaxChars`, `promptCacheTtl`, and `subagentPromptCacheTtl` from `settings.json`, and shows whether reads are confined to working directories.

— Lists all fields but not where they surface in the UI. v1.2.0

09 **Cost attribution per skill, MCP tool, and agent** NEW 60

A new Analytics tab reports spend per skill, per MCP tool, and per agent over the selected window, and the Skills, MCPs, and Agents rails each carry a lifetime spend chip.

— Describes the feature and location but no metrics or limits given. v1.2.0

THINNER COVERAGE BELOW

10 **One-time full reparse on upgrade** 50

BREAKING

First launch after upgrading to v1.2.0 triggers a one-time full reparse due to a parser version bump and a pricing-table key change; subsequent launches hydrate from cache as before.

— Explains cause and one-time impact of the upgrade behavior. v1.2.0

---

11 **Subagent model, effort level, and cost in agent tree** NEW

45

The agent tree and detail card now show the subagent's model and effort level alongside the estimated cost of the run.

— Simple UI addition with no further mechanism described. v1.2.0

└──▷ **BREAKING ON UPGRADE**

! First launch after upgrading triggers a one-time full reparse due to a parser version bump and a pricing-table key change; subsequent launches hydrate from cache as before.

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ 1 RELEASE · 2026-09-17 BLOG ↗

RANK

Amp is Sourcegraph's agentic coding tool for the terminal and editor, running multi-step edits with subagents and shared team threads.

> Amp runners can now serve multiple project directories at once and manage them dynamically without restarting, while also self-updating in the background.

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

01 **Multi-directory serving for Amp runners** NEW 95

A single runner can now serve multiple project directories at once via the repeatable `--dir` flag on `amp --no-tui`, or automatically discover and serve every Git checkout up to two levels beneath the current directory (including newly cloned repos) via `--discover-dirs`. New `amp runner dirs add`, `amp runner dirs list`, and `amp runner dirs remove` subcommands let you dynamically manage the directories a runner serves without restarting it.

Consolidate multiple projects onto one always-on machine so every repo is available to Amp without running a separate runner per project.

```
$ amp --no-tui --runner-id mac-mini --dir ~/code/amp --dir ~/code/sandcastle
```

Let a runner automatically pick up any new Git repo you clone into a workspace directory, without editing config or restarting.

```
$ cd ~/code && amp --no-tui --runner-id mac-mini --discover-dirs
```

— Names exact flags and subcommands with runnable examples

One Runner Is Now Enough (2026-09-17)

02 **Runner self-update mechanism** NEW 80

A running `amp --no-tui` process now checks for new releases roughly once per hour and restarts into the new version (at most once per 12 hours) when no thread is active. The `amp.runner.autoUpdate.enabled` setting controls whether a runner self-updates, letting teams opt long-running shared runners out of automatic upgrades.

Opt a long-running shared runner out of automatic version upgrades to keep a stable environment for the team.

```yaml
amp.runner.autoUpdate.enabled: false
```

— Names the config key and gives exact update cadence

One Runner Is Now Enough (2026-09-17)

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ 1 RELEASE · 2026-09-16 BLOG ↗ RANK

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

> Kiro rolled out Claude Fable 5.1 in Preview for Enterprise organizations, offering a 1M-token context window with a 6x credit multiplier and admin-controlled access.

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

01 Claude Fable 5.1 Preview for Enterprise NEW 75

Claude Fable 5.1, with a 1M-token context window and a 6x credit multiplier, is rolling out in Preview for Kiro Enterprise organizations. Administrators enable model access management and then add Fable 5.1 from the Manage approved list to make it available to their users; all Fable 5.1 traffic is retained for up to 30 days for automated offline abuse detection, with classifier-flagged traffic subject to potential human review by AWS.

Enable Fable 5.1 for your Kiro Enterprise org so users can leverage the 1M-token context window for large multi-file refactors.

📍 1. In the Kiro Enterprise admin console, go to Settings › Model governance and enable model access management. 2. Navigate to Manage approved list. 3. Locate Claude Fable 5.1 and add it to the approved list. 4. Users in your organization can now select Fable 5.1 as their active model in the Kiro IDE.

— Names model specs and admin steps, but no API/config key given

Models: Claude Fable 5.1 Preview Begins…

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

PydanticAI

Feedback ⓘ 1 RELEASE · 2026-09-17 NOTES ↗ RANK

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

> PydanticAI v2.44.0 stabilizes agent run result serialization for persistence and replay, adds realtime event typing and a Storage UI page, and ships two breaking changes affecting UI adapter requests and durable operation dispatch.

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

01 JSON Content-Type required for UI adapter requests

BREAKING

60

UI adapter requests now require a JSON `Content-Type` header; requests without it will be rejected.

— Clear header requirement readers can check immediately v2.44.0

02 Durable operation dispatch from per-request hooks changed

60

BREAKING

A `@durable_operation` called from a per-request hook is now dispatched as a capability instead of running inline; workflows that relied on inline execution will behave differently.

— Names decorator and behavior change, no migration steps v2.44.0

THINNER COVERAGE BELOW

03 Stable AgentRunResult serialization IMPROVED

50

`AgentRunResult` now has a stable serialized shape, and a finished stream is settled into one, enabling reliable persistence and replay of agent run results.

— Explains mechanism but no code example given v2.44.0

04 EnqueuedMessagesEvent for realtime sessions NEW

45

Emits a typed `EnqueuedMessagesEvent` when a realtime session delivers enqueued content, giving listeners a typed event for streamed realtime output.

— Names event type but no usage shown v2.44.0

05 Vercel AI SDK and Eve migration skill NEW

40

Adds a migration skill to assist developers migrating existing Vercel AI SDK and Eve projects to PydanticAI.

— Names source frameworks but no steps given v2.44.0

06 **Storage page in UI** NEW 35

Adds a `Storage` page to the UI so persistence configuration is discoverable.

— Names UI page but lacks configuration detail v2.44.0

└──▷ **BREAKING ON UPGRADE**

! UI adapter requests now require a JSON `Content-Type` header; requests without it will be rejected.

! A `@durable_operation` called from a per-request hook is now dispatched as a capability instead of running inline; workflows that relied on inline execution will behave differently.

WAS THIS USEFUL?  

CrewAI

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

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

> CrewAI 1.15.22 introduces per-role model routing with a new context variable, adds OpenRouter as an embedding provider, and rounds out several CrewAI Platform surfaces covering tracing, execution payloads, and integration setup.

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

01 **Per-role model routing via llm_overlay** NEW 60

Adds the `llm_overlay` context variable so specific agent roles can be routed to different LLMs at runtime, rather than every agent in a crew using the same model.

— Names the config variable but no usage snippet shown 1.15.22

THINNER COVERAGE BELOW

02 **Human feedback tracing and richer execution payloads** IMPROVED 50

CrewAI Platform tracing now collects human feedback and pause events, and agent execution payloads carry `task_prompt` and output fields for that tracing.

— Names payload fields but not how to inspect them 1.15.22

03 **Platform integration and deployment management improvements** IMPROVED 50

CrewAI Platform gains an exposed application catalog, platform tools added to the JSON crew wizard, validation of platform integrations during crew setup, recorded reasons for deployment creation failures, and support for aliases as connection identifiers.

— Several distinct platform surfaces named, each described thinly 1.15.22

04 **OpenRouter embedding provider support** NEW 35

OpenRouter is now a supported embedding provider for CrewAI.

— Provider named but no setup detail given 1.15.22

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ 2 RELEASES · 2026-09-16 NOTES ↗ RANK

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

> Vercel AI SDK shipped an experimental typed LLM evaluation API and a native tool-search tool for agents in this window.

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

01 Typed LLM evaluation via experimental_evaluate NEW

95

`experimental_evaluate` and an isolated experimental v4 evaluation model specification let you run typed LLM evaluations with Choice, Score, and Boolean question types against shared state, returning typed answers with optional Choice/Score distributions and required Boolean probabilities, plus validation, retries, cancellation, and `Experimental_EvaluationUnsupportedQuestionTypeError` for unsupported question types. Custom providers can now declare an `evaluationModels` option (with a matching `evaluationModel` accessor on registries) so `experimental_evaluate` can resolve evaluation models by string ID when an evaluation-capable default provider is explicitly configured, and model-resolution errors now identify `evaluationModel`.

Register evaluation models on a custom provider so `experimental_evaluate` can resolve them by string ID without falling back to Gateway.

typescript

```typescript
import { customProvider, experimental_evaluate } from 'ai';

const myProvider = customProvider({
  languageModels: { /* ... */ },
  evaluationModels: {
    'my-eval-model': someEvaluationModel,
  },
});

const result = await experimental_evaluate('my-eval-model', { /*
eval config */ });
```

— Names types, error class, config keys, and includes runnable code. ai@7.0.104 ai@7.0.103

THINNER COVERAGE BELOW

02 Native tool search tool NEW

18

Adds a native tool search tool for direct tool-calling search support.

— Only a bare description with no usage or mechanism given. `ai@7.0.104`

WAS THIS USEFUL? 👍 👎

holmesgpt

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

SRE Agent - CNCF Sandbox Project

> HolmesGPT 0.42 adds namespace-scoped RBAC installs, GPU node diagnostics for Kubernetes remediation, configurable New Relic toolset timeouts, and richer MCP tool-call handling.

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

01 Namespace-scoped RBAC Helm install option NEW 80

Adds the `namespaceScopedRBAC` Helm value to support namespace-scoped HolmesGPT installs without requiring cluster-wide permissions, useful in multi-tenant clusters where cluster-admin cannot be granted.

Install HolmesGPT scoped to a single namespace instead of requiring cluster-wide RBAC — useful in multi-tenant clusters where you cannot grant cluster-admin.

```
$ helm upgrade --install holmes robusta/holmes --set
namespaceScopedRBAC=true --namespace holmes
```

— Named Helm flag with a runnable install command 0.42.0

THINNER COVERAGE BELOW

02 MCP structured content and payload cleanup IMPROVED 35

Surfaces MCP `structuredContent` in tool responses and drops null optional parameters in MCP tool calls.

— Names the field but not how it's consumed 0.42.0

03 Conversation link field in chat context NEW 35

Adds a `conversation_link` chat field that is rendered into the system prompt for richer chat context.

— Names field and effect but no usage example 0.42.0

04 GPU node diagnostics in remediation toolset NEW 30

Adds GPU node diagnostics support to the `kubernetes-remediation` MCP toolset.

— Names the toolset but no mechanism or usage detail 0.42.0

05 Configurable New Relic toolset timeout NEW 25

Adds a configurable request timeout for the New Relic toolset.

— Bare mention, no config key or default given `0.42.0`

WAS THIS USEFUL?  

Agno (formerly Phidata)

Feedback ⓘ 1 RELEASE · 2026-09-16

NOTES ↗ RANK

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

> Agno v3.0.10 adds Azure OpenAI Responses model support, an Elasticsearch vector database backend, a Markdown transform for documentation ingestion, and new MCP hosting/routing options, alongside two breaking security defaults around shell execution and MCP access.

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

01 **Custom hostname and path routing for MCP** NEW 85

Adds `root_host`, `path`, and `path_aliases` to MCPConfig to serve MCP on a dedicated hostname or a custom endpoint path, supporting multi-tenant deployments.

Host your MCP server on a dedicated subdomain with a custom endpoint path for multi-tenant deployments.

python

```python
from agno.mcp import MCPConfig

mcp_config = MCPConfig(
    root_host="mcp.example.com",
    path="/v1/mcp",
    allowed_hosts=["mcp.example.com"],
)
```

— Exact config fields with a runnable example. v3.0.10

02 **CodingTools run_shell now opt-in** BREAKING 70

`CodingTools.run_shell` is now opt-in via `enable_run_shell=True`; restricted mode runs commands without a shell, so existing code relying on shell execution must explicitly enable it.

— Exact flag named; no migration example provided. v3.0.10

03 **MCP access restricted to localhost by default** BREAKING 70

With `authorization=True` and PublicSurface(mcp=True), MCP now accepts only localhost connections by default; add your domain via MCPConfig(allowed_hosts=[...]) to restore external access.

— Exact remediation config named, but no full example. v3.0.10

04 **Azure OpenAI Responses API support** NEW 65

Adds the `AzureOpenAIResponses` class to use the Azure OpenAI Responses API instead of the standard Chat Completions API with an Azure deployment.

Use the Azure OpenAI Responses API instead of the standard Chat Completions API with an Azure deployment.

```python
from agno.models.azure import AzureOpenAIResponses
from agno.agent import Agent

agent = Agent(
    model=AzureOpenAIResponses(),
    instructions="You are a helpful assistant.",
)
agent.print_response("Summarise the OWASP Top 10.", stream=True)
```

— Runnable example given, but no further mechanism detail. v3.0.10

05 **DocumentationMarkdown transform for Knowledge.sync_pages** NEW 60

Adds a `DocumentationMarkdown` transform for `Knowledge.sync_pages` that converts Mintlify and Fumadocs components into plain Markdown.

— Names transform and source formats, no usage example. v3.0.10

THINNER COVERAGE BELOW

06 **Elasticsearch vector database backend** NEW 50

Adds an Elasticsearch vector database backend with vector, keyword, and hybrid search support.

— Names backend and search modes but no example or config. v3.0.10

└──▷ **BREAKING ON UPGRADE**

! `CodingTools.run_shell` is now opt-in — existing code that relied on shell execution must explicitly pass `enable_run_shell=True`, and restricted mode now runs commands without a shell.

! With `authorization=True` and PublicSurface(mcp=True), MCP now accepts only localhost by default; add your domain to MCPConfig(allowed_hosts=[...]) to restore external access.

WAS THIS USEFUL? 👍 👎

Stacklok Mecatl

Feedback ⓘ 5 RELEASES · 2026-09-10 → 2026-09-15

NOTES ↗ RANK

Mecatl is an open-source agent harness for running the same provider-agnostic agent loop locally, remotely, or on Kubernetes, with durable state, permissions, and gRPC/HTTP-SSE APIs.

> Mecatl's latest releases add MCP OAuth dynamic client registration and native OIDC login for LLM endpoints, expand the `mecatui` terminal UI with connector status, navigation and conversation anchors, add Kubernetes deployment telemetry IDs, and tighten Slack bot access control with a `SLACK_ALLOWED_CHANNEL_IDS` allowlist and pluggable `AccessResolver`.

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

01 **Slack bot access control via allowlist and pluggable resolver** NEW 85

The Slack bot integration gains a `SLACK_ALLOWED_CHANNEL_IDS` environment variable to restrict which channels it will respond to via a channel-ID allowlist, and a pluggable `AccessResolver` with a default email allowlist resolver for controlling user access.

— Exact env var and resolver interface named, ready to configure. v0.0.33 v0.0.31

02 **mecatui interface and provider config improvements** IMPROVED 70

`mecatui` gained expanded local file mentions for richer in-terminal context, a live MCP connector status inventory surfacing broker status for all connected MCP connectors, readline-based prompt navigation for keyboard-driven editing, and logical conversation anchors for structured navigation within agent conversations. Provider configuration and `mecatui` commands were also unified into a single surface.

— Names every constituent addition but gives no exact commands. v0.0.37 v0.0.36 v0.0.34

THINNER COVERAGE BELOW

03 **Stable installation telemetry ID in mecak8s** NEW 55

Adds a stable installation telemetry ID to `mecak8s` deployments for consistent observability across pod replacements.

— Names component and purpose but no config key given. v0.0.34

04 **Session lifecycle surface in SDK** NEW 50

The SDK now exposes a session lifecycle surface, enabling programmatic session management for agent sessions.

— Names capability but no exact API method shown. v0.0.37

05 **Native OIDC login for LLM endpoints** NEW 45

Mecatl adds native OIDC login for LLM endpoints, so agents can authenticate to model providers using OIDC tokens.

— Explains mechanism but no flag or config key named. v0.0.36

06 **MCP OAuth dynamic client registration** NEW 40

Mecatl now supports MCP OAuth dynamic client registration directly, allowing MCP connectors to register OAuth clients without manual setup.

— Named capability but no endpoint or config detail given. v0.0.37

07 **Version logging on mecated startup** NEW 40

Logs the server version on `mecated` startup for easier runtime identification.

— Simple logging addition, no further detail. v0.0.33

08 **Opt-out OTLP product metrics** NEW 35

Adds opt-out product/adoption metrics exported over OTLP.

— OTLP named but no opt-out flag or config key given. v0.0.36

09 **Caller validation observability in auth layer** NEW 35

Adds caller validation observability to the auth layer for tracking and inspecting caller identity checks.

— No metric name or endpoint provided. v0.0.34

10 **Bash tool renamed to Shell** BREAKING 35

The built-in Bash tool has been renamed to Shell.

— Simple rename, no migration guidance given. v0.0.34

11 **Synthetic vs real prompt distinction in replay** IMPROVED 30

Session replay now distinguishes synthetic prompts from real ones.

— Thin description with no mechanism or UI location. v0.0.37

12 **Draft-aware session inventory** NEW 30

Adds draft-aware session inventory so in-progress sessions are tracked and visible.

— No UI location or command specified. v0.0.36

13 **Native Anthropic and Deno support** NEW 15

Mecatl adds native support for Anthropic and Deno, per the release summary; no further detail on configuration or usage was provided.

— Only mentioned in summary, no bullet detail. v0.0.36

WAS THIS USEFUL? 👍 👎

Brig

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

Brig runs coding agents in a microVM sandbox, made up of the brig CLI and the brigd session daemon.

> Brig v0.2.0 adds preview and confirmation flags to its removal commands, bundles hull and cosign into the installer, and drops support for Intel Macs.

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

01 **Unattended confirmation flag for rm/reset** BREAKING

85

`brig rm --all` and `brig reset` now accept a `-y` flag to confirm unattended removal without a terminal on stdin. This is a breaking change: without a terminal on stdin, either command now refuses to run unless `-y` is passed, so scripts must be updated to e.g. `brig rm --all -y`.

— Exact flags, commands, and required migration are all stated. v0.2.0

02 **Dry-run preview for brig rm** NEW

63

`brig rm` now accepts a `--dry-run` flag to preview what would be removed without deleting anything.

— Names exact flag and command but no mechanism beyond preview. v0.2.0

THINNER COVERAGE BELOW

03 **Bundled hull/cosign install, Intel Mac dropped** BREAKING

55

`brig install` now installs `hull` and `cosign` automatically as part of setup, and refuses to run on Intel Macs.

— Names the components and platform restriction but no further mechanism. v0.2.0

└──▷ **BREAKING ON UPGRADE**

! `brig rm --all` and `brig reset` without a terminal on stdin now refuse to run unless `-y` is passed; scripts that ran either unattended must be updated to `brig rm --all -y`.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Together AI

Feedback ⓘ 1 RELEASE · SEEN 2026-09-17 NOTES ↗ RANK

Run, train, and serve open-source AI models on Together AI.

> Together AI added a LiteLLM gateway integration for unified OpenAI-format access, brought GPT-OSS models onto its inference API, and shipped Prometheus/Grafana monitoring, a Hugging Face integration, a Next.js quickstart, and a new Python SDK v2.

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

01 LiteLLM gateway integration NEW

95

Adds LiteLLM (v1.100.0+) as an OpenAI-format gateway to Together AI, installed via `litellm>=1.100.0` and authenticated with the `TOGETHERAI_API_KEY` environment variable. Supports `stream=True`, `tools` (function calling), `response_format` (JSON mode), and `reasoning_effort` pass-through in OpenAI format. Reasoning models return chain-of-thought in a separate `reasoning_content` field, keeping `message.content` as the final answer only, and the proxy can be used to route Claude Code to Together AI models.

Call a Together AI model through LiteLLM's unified OpenAI-format interface — useful for teams already using LiteLLM to avoid provider-specific SDKs.

```
$ export TOGETHERAI_API_KEY=<your_key>
```

Use reasoning models via LiteLLM and read the chain-of-thought separately from the final answer.

```python
import litellm
response = litellm.completion(
    model='together_ai/zai-org/GLM-5.3',
    messages=[{'role': 'user', 'content': 'Explain quantum
entanglement'}],
    reasoning_effort='high'
)
reasoning = response.choices[0].message.reasoning_content
answer = response.choices[0].message.content
```

— Names exact package version, env var, and field behaviour with runnable code

product docs

02 **GPT-OSS model support** NEW 65

Adds `openai/gpt-oss-120b` as a supported model on Together AI's OpenAI-compatible inference API, enabling use of OpenAI's open-weight reasoning model on Together AI.

Run inference against the new 120B model using the OpenAI-compatible endpoint.

```
$ curl https://api.together.xyz/v1/chat/completions \
  -H 'Authorization: Bearer $TOGETHER_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"model": "openai/gpt-oss-120b", "messages": [{"role":
"user", "content": "Hello"}]}'
```

— Names exact model id and includes runnable curl call `product docs`

THINNER COVERAGE BELOW

03 **Prometheus and Grafana monitoring for dedicated endpoints** NEW 50

Supports scraping a metrics endpoint with Prometheus and importing an example Grafana dashboard for dedicated endpoints.

— Names the mechanism but no endpoint URL or config key `snapshot-20260917`

04 **Together Python SDK v2** NEW 45

Introduces Together Python SDK v2 with improved type safety and modern architecture, with a migration guide from v1.

— States what changed but no install command or migration specifics `snapshot-20260917`

05 **Next.js quickstart guide** NEW 35

Adds a Next.js quickstart for building single-question and chat applications against Together AI LLMs.

— Describes the guide's purpose but no code or path given `product docs`

06 **Hugging Face inference integration** NEW 30

Adds Hugging Face inference integration, enabling Together models to be used through the Hugging Face inference API.

— Names the integration but no endpoint, config, or example given `product docs`

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ

3 RELEASES · 2026-09-16 → 2026-09-17 NOTES ↗ RANK

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

> HyperFrames added engine-level shutter-integral motion blur and a new HLS VOD output format, alongside locked-GOP streaming control, expanded render telemetry fields, and a new lint warning for scaffold size mismatches.

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

01 Shutter-integral motion blur in render engine NEW

88

Adds `motionBlur` key to render configuration, accepting `samplesPerFrame` (up to 64), `shutterAngle` (default 180°), and `shutterPhase` (default -90°) to enable sub-frame multi-sample blur; when active it forces PNG frame capture for the entire render. The blur integrates the full shutter window across all animated properties — rotation, scale, opacity, filters, and child animation — not translation alone; a static frame still costs one capture.

Enable shutter-integral motion blur with 32 samples per frame for a cinematic render that blurs all animated properties, not just translation.

```json
{
  "motionBlur": {
    "samplesPerFrame": 32,
    "shutterAngle": 180,
    "shutterPhase": -90
  }
}
```

— Config keys, defaults, mechanism and a runnable example given v0.8.45

02 Locked-GOP control for streaming encoder NEW

75

Adds `lockGopForChunkConcat` and `gopSize` to the streaming encoder, forcing keyframes at a fixed interval so encoded output can be cut and joined without re-encoding — matching the behaviour already available on the disk encoder.

— Named flags and mechanism, no example run v0.8.44

03 Root dimension mismatch lint warning NEW

73

Adds `root_dimensions_mismatch` warning to `hyperframes lint` and `hyperframes check` that fires when the root's `data-width` / `data-height` disagree with the `html, body` CSS size or the viewport meta.

— Named commands and trigger condition, no example run v0.8.43

04 **Render telemetry event field enrichment** IMPROVED 70

Extends `render_complete` telemetry events with the requested output shape, deterministic adapter usage, element counts (audio, image, sub-composition, audio group, color grading), LUT presence, browser and FFmpeg major versions, plus new `composition_element_tags`, `aroll_video_count`, and `heygen_video_count` fields for per-tag element breakdowns. Extends `render_error` events with `error_name` and `failed_stage_code` fields so failures can be grouped by error type and pipeline stage.

— All named telemetry fields listed, no usage example v0.8.43 v0.8.45

THINNER COVERAGE BELOW

05 **HLS VOD output format** NEW 59

Adds HLS VOD package as a new output format available from both the Producer and CLI — H.264/AAC content copied into MPEG-TS segments with playlists, every segment keyframe-aligned.

— Named format and pipelines but no exact command shown v0.8.45

WAS THIS USEFUL?  

Google Gen AI JavaScript SDK

Feedback ⓘ

1 RELEASE · 2026-09-16 NOTES ↗ RANK

The Google Gen AI JavaScript SDK provides TypeScript and JavaScript interfaces to Gemini APIs and live model sessions.

> The Gen AI JavaScript SDK adds a pluggable custom fetch transport, new retrieval step types in the interactions schema, a credentials management resource, and environment file upload/download support.

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

01 **Custom fetch transport via HttpOptions** NEW 75

The SDK now supports a `custom` fetch function via `HttpOptions`, allowing callers to intercept or replace the SDK's default HTTP transport.

— Names exact config option but no usage example v2.23.0

02 **Retrieval step types in interactions schema** NEW 60

Adds `RetrievalCallStep` and `RetrievalResultStep` types to the interactions schema, exposing retrieval step data in agent responses.

— Named types given but no usage guidance v2.23.0

THINNER COVERAGE BELOW

03 **Credential APIs resource** NEW 30

Adds a credential APIs resource, enabling management of credentials through the SDK.

— No named endpoints, methods, or fields given v2.23.0

04 **Environment file upload and download** NEW 30

Implements environment file upload and download across the SDK.

— No API name or usage details provided v2.23.0

WAS THIS USEFUL? 👍 👎

Google Gen AI Python SDK

Feedback ⓘ

1 RELEASE · 2026-09-16 NOTES ↗ RANK

The Google Gen AI Python SDK provides Python interfaces to Gemini model generation, streaming, tools, and embeddings.

> Google Gen AI Python SDK v2.24.0 adds structured retrieval-step tracking, a credential management resource, and environment file transfer support.

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

01 **Retrieval step types in interactions schema** NEW 45

Adds `RetrievalCallStep` and `RetrievalResultStep` types to the interactions schema, enabling structured tracking of retrieval steps in agent interactions.

— Names the two new schema types but no usage detail v2.24.0

02 **Credential APIs resource** NEW 30

Adds a credential APIs resource and wires it to the SDK, enabling programmatic management of credentials.

— Names the resource but no methods or fields given v2.24.0

03 **Environment file upload and download** NEW 20

Adds environment file upload and download support across the SDK.

— No API names or mechanism specified v2.24.0

WAS THIS USEFUL? 👍 👎

DeepSpeed

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

DeepSpeed is a deep learning optimization library for distributed training and inference, providing ZeRO sharding, pipeline parallelism, and mixed-precision execution.

> DeepSpeed v0.19.7 adds ARM CPU Adam kernels, an opt-in DeepEP transport for AutoEP, a continuous-batching rollout prototype, and reworks AutoTP to support per-model sharding metadata alongside several smaller launcher and checkpoint tweaks.

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

01 **Per-model AutoTP sharding metadata** IMPROVED 65

Replaces AutoTP `tp_shard` process-wide globals with per-model `AutoTPMeta`, enabling multiple models to be tensor-parallel sharded within a single process.

— Names old and new internal structures with before/after behavior v0.19.7

02 **Opt-in LR range test staircase flag** BREAKING 60

Makes `--lr_range_test_staircase` an opt-in flag rather than always-on.

— Exact flag named with clear before/after behavior v0.19.7

THINNER COVERAGE BELOW

03 **SLURM launcher include/exclude flags** NEW 55

Honors `--include` / `--exclude` flags in the SLURM launcher.

— Exact flags named and directly runnable v0.19.7

04 **AutoEP expert-parallelism enhancements** NEW 50

Adds opt-in DeepEP transport for the AutoEP expert all-to-all communication path, opt-in fused weighted restore for AutoEP, and partitions AutoEP expert parameters per layer under ZeRO-3.

— Names DeepEP, AutoEP, ZeRO-3 but no usage steps given v0.19.7

05 **ARM CPU Adam optimizer kernels** NEW 45

Adds an ARM SVE update kernel for the CPU Adam optimizer and a NEON SIMD path for CPU Adam on AArch64.

— Names SVE, NEON, AArch64 but no enablement steps v0.19.7

06 **Static KV cache preallocation with head_dim** IMPROVED 45

Preallocates the static KV cache with `config.head_dim`.

— Names config field but limited mechanism detail v0.19.7

07 **Configurable dtype for ZeRO checkpoint export** IMPROVED 35

Adds configurable dtype for ZeRO checkpoint export.

— Names the capability but not the config key v0.19.7

08 **DeepCompile ZeRO-3 memory scheduler split** IMPROVED 35

Splits the DeepCompile ZeRO-3 memory scheduler for finer-grained control.

— States mechanism briefly, no concrete controls named v0.19.7

09 **Affine-map export for AutoTP checkpoint shards** NEW 35

Emits affine maps from AutoTP layers, describing universal checkpoint shards as affine maps.

— Mechanism named but no usage path given v0.19.7

10 **Continuous batching in rollout path** NEW 30

Adds a continuous batching generation prototype and continuous batching profiling to the rollout path.

— Described as prototype with no usage detail v0.19.7

11 **Qwen3.5 RMSNorm recognition in AutoTP** NEW 30

Recognizes Qwen3.5 RMSNorm variants in AutoTP module loading.

— Names model and module but no further detail v0.19.7

12 **macOS MPS accelerator support** IMPROVED 30

Adds a macOS (MPS) CI workflow and a torch floor check for the MPS accelerator.

— Names MPS but describes only CI/version check v0.19.7

WAS THIS USEFUL?  

Intel OpenVINO

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗

RANK

OpenVINO optimizes and deploys AI inference across Intel CPUs, GPUs, and NPUs, with model conversion, quantization, and a generative AI runtime.

> OpenVINO 2026.4 adds Node.js speech-recognition support, two new speculative-decoding techniques for GenAI, full ITT/VTune profiling on NPUs, and support for dozens of new models across CPU, GPU and NPU, while removing a long list of deprecated Python, GenAI, NNCF and Model Server APIs and raising the CPU plugin's minimum requirement to AVX2.

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

01 Python API cleanup: removed deprecated namespaces

BREAKING

85

The deprecated `openvino.runtime` namespace has been removed (use the `openvino` namespace directly); `openvino.Type.undefined` has been removed (use `openvino.Type.dynamic` instead); and the `PostponedConstant` constructor signature has changed from `Callable[[Tensor], None]` to `Callable[[], Tensor]`.

— Exact before/after names and migration path for each API. 2026.4.0

02 NNCF deprecated APIs and TensorFlow backend removed

BREAKING

85

create_compressed_model() has been removed from NNCF; use nncf.prune() for unstructured pruning and nncf.quantize() for INT8 quantization instead. NNCF optimization methods for TensorFlow models and the TensorFlow backend have been removed, along with the experimental NAS, Structural Pruning, AutoML, Knowledge Distillation, Mixed-Precision Quantization, and Movement Sparsity methods.

— Gives exact replacement functions and enumerates every removed method. 2026.4.0

03 OpenVINO GenAI deprecated APIs removed

BREAKING

75

Removes the deprecated OpenVINO GenAI predefined generation configs; removes support for the Whisper stateless decoder model (use a stateful model instead); and removes the deprecated `StreamerBase.put` method, the `bool` return type for callbacks, and the `ChunkStreamer` class.

— Names each removed API with a migration note for one. 2026.4.0

04 **Model Server TFS API and stateful models removed** BREAKING

70

The TensorFlow Serving (TFS) API in OpenVINO Model Server has been dropped (removed in 2026.3); use the KServe API instead. Support for Stateful Models in OpenVINO Model Server has also been removed (removed in 2026.3).

— Names both removed surfaces and the replacement API. 2026.4.0

05 **ASRPipeline for Node.js in GenAI** NEW

65

Adds `ASRPipeline` to OpenVINO GenAI for Node.js, letting JavaScript developers run automatic speech recognition models (Whisper, Qwen3-ASR) with streaming and performance metrics, via a pipeline API consistent with the existing C++ and Python APIs.

— Names the API and models but no code sample given. 2026.4.0

06 **Multi-Token Prediction speculative decoding** NEW

60

Adds Multi-Token Prediction (MTP) speculative decoding for Gemma 4, Qwen3.5, and Qwen3.6 on CPUs and GPUs to increase throughput and lower latency without accuracy loss.

— Names technique, models and platforms, no usage command. 2026.4.0

07 **New model support across CPU, GPU and NPU** NEW

60

Adds support for new models on CPU: `Gemma-3n`. On CPU and GPU: `Kokoro-82M`, `Qwen3-VL-4B with EAGLE-3`, `Qwen3-ASR`, `Muse Glimmer 30B`, `Qwen3.8 27B`, `Gemma 4 12B`, `Hy-MT2-1.8B`, `DeepSeek OCR-2`, and `Granite 4.0 H Micro`. On NPU: `FLUX.2-Klein 4B` and `Kokoro-82M`.

— Every model named per platform, but no mechanism beyond the list. 2026.4.0

THINNER COVERAGE BELOW

08 **Tree Drafting for EAGLE-3** NEW

55

Adds Tree Drafting (Top-K) for EAGLE-3 in OpenVINO GenAI, enabling higher throughput on VLM pipelines compared to the previous Chain Drafting (Top-1) approach.

— Explains mechanism versus prior approach but no command shown. 2026.4.0

09 **ITT/VTune profiling extended to NPU** IMPROVED

50

Extends Instrumentation and Tracing Technology (ITT) profiling support to the NPU, allowing Intel VTune Profiler to analyze CPU, GPU, and NPU execution through one consistent toolchain.

— Names the tools and scope but no setup steps given. 2026.4.0

10 **Preview DFlash acceleration and visual-token compression** NEW

50

Adds preview support for DFlash acceleration for Qwen models on GPUs, and preview visual-token compression to reduce latency on Intel Core Ultra Series 3 processors.

– Names the mechanisms and hardware but marked preview with no usage detail. 2026.4.0

---

11 **Xe3 integrated graphics optimizations for Gemma 4** IMPROVED

50

Adds Xe3 integrated graphics optimizations in OpenVINO GenAI improving inference performance for Gemma 4 models on long-context inputs on Intel Core Ultra Series 3 processors.

– Names hardware and model but no benchmark or config given. 2026.4.0

---

12 **CPU plugin now requires AVX2** BREAKING

50

The CPU plugin now requires AVX2 as a minimum instruction set; SSE is no longer supported.

– States the new hardware floor but no migration guidance. 2026.4.0

---

13 **Bounded dynamic-shape support on NPU for vision** NEW

45

Adds preview support for bounded dynamic-shape on NPUs extended to vision workloads such as image super-resolution, validated with the ESPCN model.

– Names the validation model but no enablement steps. 2026.4.0

---

14 **Idle model unloading in Model Server** NEW

45

Adds preview idle model management to OpenVINO Model Server, unloading models when not in use to reduce memory consumption.

– Describes behavior but no config key for enabling it. 2026.4.0

---

15 **Debian 10 support discontinued** DEPRECATED

25

Support for Debian 10 has been discontinued.

– One-line platform drop with no further detail. 2026.4.0

└──▷ BREAKING ON UPGRADE

! The deprecated `openvino.runtime` namespace has been removed; use the `openvino` namespace directly.

! The deprecated `openvino.Type.undefined` has been removed; use `openvino.Type.dynamic` instead.

! The `PostponedConstant` constructor signature has changed: the old `Callable[[Tensor], None]` form is removed; the new form is `Callable[[], Tensor]`.

- ! The deprecated OpenVINO GenAI predefined generation configs have been removed.
- ! The deprecated OpenVINO GenAI support for the Whisper stateless decoder model has been removed; use a stateful model instead.
- ! The deprecated OpenVINO GenAI `StreamerBase.put` method, `bool` return type for callbacks, and `ChunkStreamer` class have been removed.
- ! NNCF create_compressed_model() has been removed; use nncf.prune() for unstructured pruning and nncf.quantize() for INT8 quantization.
- ! NNCF optimization methods for TensorFlow models and the TensorFlow backend in NNCF have been removed.
- ! The following experimental NNCF methods have been removed: NAS, Structural Pruning, AutoML, Knowledge Distillation, Mixed-Precision Quantization, Movement Sparsity.
- ! The CPU plugin now requires AVX2 as a minimum instruction set; SSE is no longer supported.
- ! Support for Debian 10 has been discontinued.
- ! The TensorFlow Serving (TFS) API in OpenVINO Model Server has been dropped (removed in 2026.3); use the KServe API instead.
- ! Support for Stateful Models in OpenVINO Model Server has been removed (removed in 2026.3).

WAS THIS USEFUL? 👍 👎

PyTorch ExecuTorch

Feedback ⓘ 1 RELEASE · 2026-09-16

NOTES ↗ RANK

ExecuTorch runs PyTorch models on mobile, embedded, and edge devices through an ahead-of-time export path and a lightweight C++ runtime.

> ExecuTorch v1.5.0 adds a C++ SDK with linkable delegate libraries and CUDA wheel publishing, opt-in ROCm support for the CUDA backend, multi-method export, and broad backend expansions across Arm/Ethos-U, Qualcomm, Apple (Core ML/MLX/Metal), Vulkan, and NXP Neutron, plus new quantized LLM/MoE operators and serving workflows.

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

01 **Arm and Ethos-U backend expansion** NEW 80

Adds Arm export-recipe support, a pre-decomposition partitioner pipeline, and broader symbolic-shape and dim-order handling; adds Ethos-U65 support and Ethos-U85 KV-cache export with SmolLM2, TinyStories-42M, Silero VAD, MobileSAM, and Stable Diffusion workflows; and expands Arm VGF profiling and environment preflight validation tools with int8 KV cache, FP16, BF16, MXFP8, and quantized model coverage. Install with `pip install 'executorch[ethos_u]'` to access the Ethos-U65/U85 targets.

Install ExecuTorch with Ethos-U AOT export dependencies to access new Ethos-U65 and Ethos-U85 targets.

```
$ pip install 'executorch[ethos_u]'
```

— Many named targets and workflows plus an install command v1.5.0

02 **CUDA and ROCm backend improvements** IMPROVED 70

Adds opt-in ROCm support to the AOTI CUDA backend, including packed W4 execution and Voxtral Realtime coverage on `gfx1100` and `gfx950`; adds shared-memory targeting for cross-architecture CUDA inference and INT4, INT5, and INT6 short-query kernels; and improves the CUDA delegate with per-thread streams, a warm memory pool, and cross-method weight sharing.

— Multiple named targets and kernels but no runnable command v1.5.0

03 **Apple Core ML, MLX, and Metal backend expansion** NEW 70

Adds linkable Core ML and MLX libraries to Apple frameworks and SwiftPM, plus ETDump profiling support; adds MLX off-graph KV-cache flat, ring, and cell layouts with shared pools and expands MLX lowering with fused attention, `native_group_norm`, `upsample_nearest2d`, `flip`, and `trunc` support; and adds Metal `addmm` and `split_copy` support.

— Names ops and libraries across three Apple backends v1.5.0

04 **LLM/MoE quantized operators and serving workflows** NEW

70

Adds a quantized MoE feed-forward operator with INT4 packing, fused projections, and source transformations; adds a quantized stacked-halves RoPE operator and BF16-by-FP32 GEMV decode support; adds Qwen3.5 MoE, Muse Glimmer, Supertonic, and Voxtral LLM serving workflows; and adds batched request scheduling and bounded cancellation to LLM serving.

— Names operators and workflows but no runnable example v1.5.0

05 **Qualcomm backend expansion** NEW

65

Adds LLM quantization-aware distillation and Gemma 4 text-model enablement for the Qualcomm backend, plus Qualcomm support for `pdist`, `sort`, `reflection_pad3d`, `as_strided`, `scatter_reduce`, `scatter_add`, and Hadamard-transform flows.

— Names ops and models but no runnable example v1.5.0

06 **Vulkan backend expansion** NEW

65

Adds Vulkan external image and buffer wrapping with caller-provided dimensions; adds Vulkan device-safe dispatch geometry and two-dimensional dispatch helpers; adds quantized convolution paths using unsigned dot products; and adds payload-bounded constant sharding and scalar comparisons, with broader `arange`, `clamp`, indexing, and batch support.

— Detailed mechanism but no API entry point or command v1.5.0

07 **NXP eIQ Neutron backend expansion** IMPROVED

65

Upgrades the NXP eIQ Neutron SDK to `3.2.1` and adds an MCUXpresso example build, plus Neutron support for `rsqrt`, Hardswish, `softmax`, reflected padding, scalar elementary arguments, and AdaptiveAvgPool1d.

— Named SDK version and ops but no command shown v1.5.0

08 **.pte/.ptd schema-version checks** IMPROVED

60

Adds schema-version checks for `.pte` and `.ptd` files with clearer failures for invalid device-planned copies and missing delegate data.

— Names file formats and failure modes but no command v1.5.0

09 **XNNPACK BF16 support** NEW 60

Adds BF16 fully connected delegation and BF16 activation support in LLM export for XNNPACK.

Export a Llama 3.2 model targeting the XNNPACK backend with BF16 activation support, now available in v1.5.0.

```
$ python -m executorch.extension.llm.export.export_llm --model
llama3_2 --output llama.pte
```

— Ties to a runnable export command for Llama 3.2 v1.5.0

THINNER COVERAGE BELOW

10 **C++ SDK and CUDA wheel packaging** NEW 55

Adds a C++ SDK to the Python wheel with linkable libraries for the CUDA, Core ML, MLX, OpenVINO, and Qualcomm delegates, plus packaged TorchAO and quantized-kernel libraries, and adds CUDA wheel publishing on pytorch.org.

— Names delegates and libraries but no install command shown v1.5.0

11 **Multi-method model export** NEW 45

Adds multi-method model export support to the export pipeline with per-backend load options.

— Describes capability without exact API or flag v1.5.0

12 **Build system and platform support** IMPROVED 45

Adds Windows ARM64 build coverage and lowers the supported CMake floor so builds work with CMake `3.26` through `3.28`.

— Concrete version range but thin on mechanism v1.5.0

13 **Capacity-aware memory planning** NEW 40

Adds capacity-aware memory planning over target memory maps and support for shared allocations with offsets.

— Mechanism named but no configuration surface given v1.5.0

14 **Shared pre-decomposition and graph-transform hooks** NEW 40

Adds shared pre-decomposition and graph-transform hooks, including common view/permute propagation and reinplace passes.

— Names passes but no API entry point v1.5.0

15 **Cortex-M explicit-layout lowering** NEW 40

Adds opt-in Cortex-M explicit-layout lowering with convolution and pooling kernels.

— Names kernel types but no flag or command v1.5.0

WAS THIS USEFUL? 👍 👎

Baseten

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

Baseten deploys and serves machine learning models as autoscaling inference endpoints, with a model API catalog and dedicated GPU deployments.

> Baseten added server-side tool execution to its Model APIs, letting the platform run tools like web search inside a single API request instead of requiring client-side dispatch.

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

01 **Server-side web search and tool execution in Model APIs** NEW 70

Baseten Model APIs now support server-side tool execution, where Baseten runs the selected tool and returns its result to the model within the same API request — no client-side tool dispatch needed. This includes built-in web search, delivered through Baseten Grounded Inference and Baseten Hosted Tools, with support for Exa, Keenable, Parallel, and You.com as search providers; the capability is in early access on enabled workspaces and models, with protocol and continuation examples in the web search guide.

— Explains mechanism and named providers but no request syntax given.

Web search with Baseten Hosted Tools

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

MLX Serve

Feedback ⓘ 1 RELEASE · 2026-09-17 NOTES ↗ RANK

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

> mlx-serve v26.9.4 adds a greedy-decoding shortcut, deeper `/props` introspection, an in-app benchmark tool, stricter image-input validation, and higher concurrency on Qwen3.8 Flash Next.

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

01 Expanded `/props` serving-settings inspection IMPROVED

85

The `/props` endpoint now reports active serving settings for the loaded model, including KV quantization, MTP, drafter, PLD, attention quantization, and prefill chunking.

Inspect all active serving settings for the currently loaded model — useful for confirming KV quantization and speculative decoding configuration without restarting the server.

```
$ curl http://localhost:8080/props
```

— Endpoint and all reported settings named, with runnable curl example v26.9.4

02 Greedy decoding via `top_p: 0` NEW

75

Adds `top_p: 0` as a greedy-decoding shorthand, equivalent to `top_k: 1`, which selects only the highest-probability token without needing to set `top_k` directly.

Force fully deterministic, greedy output in a completion request without setting `top_k`, using the new `top_p: 0` shorthand.

```
$ curl http://localhost:8080/v1/chat/completions -H 'Content-Type: application/json' -d '{"model": "qwen3.8", "top_p": 0, "messages": [{"role": "user", "content": "Explain TLS handshake"}]}'
```

— Names exact parameter and equivalence; includes runnable curl request v26.9.4

03 Menu-bar model benchmark NEW

63

Adds a menu-bar benchmark that runs a standardized context-and-coding benchmark against the loaded model using the server's own timings, with optional anonymous sharing of results to mlxserve.com/benchmarks.

.

Run the standardized benchmark against the loaded model and review local results or opt in to anonymous community sharing.

> 📍 Click the mlx-serve menu-bar icon, then select `Benchmark` to start a standardized context-and-coding benchmark against the loaded model. Results appear locally; toggle anonymous sharing to contribute to mlxserve.com/benchmarks.

— Describes benchmark and sharing destination; UI path given, not a command v26.9.4

THINNER COVERAGE BELOW

04 **Strict 400 error on invalid image input** BREAKING 45

Invalid images — unreadable files, bad base64, or unsupported payloads — now return a `400` error instead of silently being dropped from the prompt.

— Clear before/after behavior but no example or endpoint given v26.9.4

WAS THIS USEFUL? 👍 👎

vMLX

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

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

> vMLX 1.6.61 adds support for a new model family and tightens concurrency, caching, and tool-result handling across the runtime.

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

01 **Persistent SSD cache validation** IMPROVED 70

Improves persistent SSD caching to validate explicit quantization metadata, geometry, and cache chains before restoring affine-quantized entries, treating model weight precision separately from cache precision.

— Clear mechanism and scope, no config surface named v1.6.61

THINNER COVERAGE BELOW

02 **GLM-5.3 Flash model support** NEW 55

Adds support for the GLM-5.3 Flash model, queuing concurrent clients behind the native runtime's single active sequence. Server settings now expose effective batch values, and health/cache statistics expose the admission policy.

— Names model and exposed stats but no exact flag names v1.6.61

03 **Qwen concurrent decoding improvements** IMPROVED 50

Improves Qwen concurrent decoding to preserve each request's positions and sparse masks across unequal-length batches, completed-request removal, and new continuations, with request-specific native SSD tool continuations.

— Detailed mechanism but no runnable surface named v1.6.61

04 **Cache telemetry accuracy fix** IMPROVED 45

Improves cache telemetry to distinguish resident native allocation from logical snapshot bytes, no longer counting spare capacity as serialized state.

— Explains behavior change but not how to observe it v1.6.61

05 **MCP tool-result continuation handling** IMPROVED 40

Improves MCP tool-result continuations to preserve detailed MCP errors and retain a model's native default reasoning effort when the default is selected.

— Describes fix but no reproduction steps or config v1.6.61

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

okf-memory OKF Agent Memory

Feedback ⓘ

2 RELEASES · 2026-09-17 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 added Git-native encrypted sync between agents and a remote hub, backed by a new zero-knowledge vault encryption layer and Bearer Token authentication for hub commands.

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

01 Hub sync commands for encrypted remote knowledge NEW

95

Adds `pkg/sync`, a low-overhead push/pull reconciliation protocol against blind CAS backends, exposed through new `okf hub` subcommands: `okf hub push` performs client-side encrypted delta sync via CAS against a remote hub using `/blobs/check-missing` with atomic head pointer updates; `okf hub pull` atomically clones or pulls the latest encrypted remote tree and decrypts it locally to disk; `okf hub sync` runs automated two-way sync with 3-way reconciliation and non-destructive conflict preservation, writing conflicts to `<file>.conflict-local.md`; and `okf hub serve` runs a built-in blind CAS and atomic head pointer server for self-hosting or zero-dependency local testing.

Run a self-hosted blind CAS hub locally to test encrypted sync without any external service.

```
$ okf hub serve
```

— Enumerates every subcommand, endpoint, and file it touches. v0.4.0 v0.4.0-rc.1

02 Zero-knowledge vault encryption for knowledge bundles NEW

83

Adds `pkg/vault`, providing zero-knowledge bundle encryption using Argon2id key derivation and AES-256-GCM authenticated envelopes over content-addressed asset trees. The new `okf hub init-vault` command scaffolds these credentials, generating a Vault ID, a 128-bit Secret Key Emergency Kit, and a `.okf-vault.json` file.

Bootstrap an encrypted vault before syncing a knowledge bundle to a remote hub for the first time.

```
$ okf hub init-vault
```

▶ View command output v0.4.0

— Names encryption mechanism and the vault scaffolding command. v0.4.0-rc.1 v0.4.0

---

03 **Bearer token authentication for hub commands** NEW 70

Adds Bearer Token authentication for `okf hub` commands via the `-auth-token` flag, the `OKF_HUB_TOKEN` environment variable, or an `auth_token` field stored in `.okf-vault.json`.

Push local knowledge changes to a token-protected hub, authenticating without storing credentials in `.okf-vault.json`.

```
$ okf hub push -auth-token <token>
```

— Names all three credential surfaces but little else. v0.4.0

WAS THIS USEFUL? 👍 👎

timgordontg engrim

Feedback ⓘ 1 RELEASE · 2026-09-17

NOTES ↗ RANK

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

> engrim v1.4.7 adds an MCP result-size hint so Claude Code keeps the engrim boot pack inline in model context instead of offloading it to a disk preview.

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

01 **MCP result-size hint for Claude Code context** IMPROVED

73

Adds `_meta: {"anthropic/maxResultSizeChars": 120000}` to the `engrim_context` MCP tool result so Claude Code retains the boot pack in model context rather than offloading it to a disk file preview.

— Names exact meta key and value but no broader mechanism detail. v1.4.7

WAS THIS USEFUL? 👍 👎

GrowGraph OntoCast

Feedback ⓘ 1 RELEASE · 2026-09-16 CODE ↗

RANK

OntoCast generates RDF knowledge graphs from documents by co-evolving domain ontologies and fact graphs with an agentic LLM pipeline.

> OntoCast's densest window yet adds an insert-only numeric-measurement completion pass and extensive critic-loop safety limits, alongside dozens of new environment-variable controls spanning chunking, retrieval windowing, LLM request pacing and prefix-caching, ontology chapter formatting, and SHACL shapes prompting — plus a breaking removal of the `CrossQueryMergeMode.HYBRID` enum value.

└──▷ **WHAT SHIPPED · 21 FEATURES** most completely described first ? what's the number?

01 **Critic loop pass budgets and safety limits** NEW 96

`FACTS_CRITIC_PASSES` (default 1) sets critic review-and-patch calls per facts unit, stopping early when a pass changes nothing (`0` =exactly one provider call per unit). `ONTOLOGY_CRITIC_PASSES` (default 0, previously unreachable at defaults) enables a separate ontology critic loop that self-applies fixes, stops early on no-op passes, and scopes deletes to the unit's own delta so retrieved catalog content is never removed. `FACTS_CRITIC_MAX_DELETE_SHARE` and `FACTS_CRITIC_MIN_DELETES` limit the fraction of triples a patch may delete, and `FACTS_CRITIC_ALLOW_SUBJECT_RENAME` (disabled by default) opts in to patches that replace a subject, avoiding orphaned nodes by default. `ONTOLOGY_ACCEPT_BLOCKING_FINDING_KINDS` configures which deterministic ontology findings block unit acceptance. Fixes compile into a `GraphUpdate` with delete-share rollback and no-op dropping, surfaced via `facts_critic_fixes_applied`, `facts_critic_fixes_applied_residual`, `facts_critic_fixes_applied_noop`, `facts_critic_patches_rolled_back`, `fix_action_severity_histogram`, and `accept_reason_histogram`.

Enable the ontology critic pass to catch and self-fix bad terms before they propagate into downstream facts — worth the one extra LLM call per unit at the ontology stage.

```
$ ONTOLOGY_CRITIC_PASSES=1 ontocast ./docs --ontology-dir ./seed
--ontology-output-dir ./ontology-out
```

Enable ontology critique passes (opt-in, previously unreachable at defaults) and cap how much a single pass may delete to avoid destructive runs.

```ini
ONTOLOGY_CRITIC_PASSES=2
FACTS_CRITIC_PASSES=3
FACTS_CRITIC_MAX_DELETE_SHARE=0.2
FACTS_CRITIC_ALLOW_SUBJECT_RENAME=false
```

Limit critic retries to 2 passes to keep costs predictable in automated pipelines.

```
$ export FACTS_CRITIC_PASSES=2
ontocast --ontology-dir ./ontologies/my-domain
```

— Every knob, default, and counter named with runnable examples v0.6.4

02 **Fan-out warmup and prefix-cache routing** NEW 93

`FANOUT_WARMUP_UNITS` (default `0`) runs that many units to completion before fanning out the rest, seeding the provider prefix cache so remaining units hit it instead of missing on simultaneous arrival. `LLM_PROMPT_CACHE_KEY` (OpenAI) gives a stable routing hint preventing a wide fan-out from spreading across machines with independent cache entries. `ONTOLOGY_CONTEXT_SCOPE` (`unit|document`) chooses whether the ontology chapter is generated once per unit or shared across the document — `document` is prefix-cache friendly and falls back to per-unit on empty union. `budget.prefix_cache_hit_rate` in the run manifest measures whether prefix caching is actually serving hits, and the manifest's `prompting.llm` block gains `prompt_cache_key`, `max_inflight`, `fanout_warmup_units`, and `ontology_context_scope` fields.

Warm the provider prefix cache on a large document so all but the first unit's render calls hit a shared ontology chapter, then verify the cache is actually being served before keeping the settings.

```
$ ONTOLOGY_CONTEXT_SCOPE=document FANOUT_WARMUP_UNITS=1
LLM_PROMPT_CACHE_KEY=my-corpus-v1 ontocast run <pipeline>
```

Maximise prefix-cache reuse on OpenAI by giving every unit the full document-level context, warming the cache with the first unit before the rest fan out.

```
$ ONTOLOGY_CONTEXT_SCOPE=document FANOUT_WARMUP_UNITS=1
LLM_PROMPT_CACHE_KEY=my-doc-run ontocast ./docs --ontology-dir
./seed --ontology-output-dir ./ontology-out
```

Process a corpus with document-scoped ontology context and prefix-cache warmup to reduce LLM token costs on large documents.

```
$ ONTOLOGY_CONTEXT_SCOPE=document FANOUT_WARMUP_UNITS=4
LLM_PROMPT_CACHE_KEY=my-shard-key ontocast process --ontology-dir
./ontologies --shapes-dir ./shapes corpus/
```

— Mechanism, defaults, and a verification manifest field all named `v0.6.4`

03 **Insert-only numeric measurement completion pass** NEW

90

A new complete_facts(state, atomic, inventory) async function in `ontocast/agent/complete_facts.py` proposes ADD-only `TripleFix` fixes to recover measurements the render missed, short-circuiting when `inventory.measurements` is empty. `FACTS_COMPLETION_PASSES` enables the pass to run after the critic loop, and `FACTS_NUMERIC_COVERAGE_LIMIT` (default 30) with `FACTS_NUMERIC_COVERAGE_MANDATORY` (default off) enforces that every stated measurement is present. A new `UNIT_SYMBOL_CASE_MISMATCH` finding flags value nodes whose unit symbol matches text only after case folding when a different catalog unit declares the exact form. The manifest's `completion` block records `measurements_recovered`, `subjects_inserted`, `subjects_rolled_back`, and `triples_inserted`, all zero when `FACTS_COMPLETION_PASSES=0`.

Recover numeric measurements missed by the render pass for a given unit, wiring the new completion agent into an existing facts pipeline.

python

```python
from ontocast.agent.complete_facts import complete_facts

# inventory is pre-computed so the completion pass and
NUMERIC_COVERAGE agree
fixes = await complete_facts(
    state=unit_facts_state,
    atomic=atomic_toolbox,
    inventory=numeric_inventory,
)
# fixes contains ADD-only TripleFix proposals; apply via
compile_critic_fixes / _apply_patches
```

— Full mechanism, defaults, and manifest fields named with a code example `v0.6.4`

04 **Chunk sizing, non-content routing, and semantic chunking** NEW

90

`CHUNK_MIN_UNIT_CHARS` drops chunks below a minimum character length, logging each as `undersized` and counting it in `undersized_units_skipped`. `CHUNK_NON_CONTENT_MODE` routes front/back matter via `skip` (increments `non_content_units_skipped`), `kept, flagged`, or `extract` (units carry `is_non_content` instead of being dropped), logging each decision with the chunk's first line. `CHUNK_MAX_MEASUREMENTS_PER_UNIT` recursively splits units exceeding a cap on unit-adjacent numbers (`0`=off). A semantic chunking mode uses clustering-based chunk boundaries (requires `torch`; falls back to paragraph/sentence splits when absent). The manifest's selection block gains `bibliography_units_skipped`, `undersized_units_skipped`, `non_content_units_skipped`, `selection.labeled_units`, `selection.unlabeled_units`, `selection.section_label_histogram`, and `selection.non_content_mode`.

Route front/back matter (author info, licence, notes) to extract mode so non-content units are flagged rather than silently dropped, then split measurement-dense units at sentence boundaries.

```ini
CHUNK_NON_CONTENT_MODE=extract
CHUNK_MAX_MEASUREMENTS_PER_UNIT=5
CHUNK_MIN_UNIT_CHARS=200
```

— Every routing knob and manifest counter named with an example `product docs`

---

05 **Retrieval proposition windowing and hybrid search tuning** NEW

90

`VECTOR_STORE_PROPOSITION_WINDOW_MAX_CHARS` bounds query windows by character length instead of sentence count, preventing encoder truncation by construction; `VECTOR_STORE_PROPOSITION_WINDOW_MAX_TOKENS` bounds windows in encoder word-piece units and hard-cuts overlong sentences at a whitespace boundary; `VECTOR_STORE_PROPOSITION_WINDOW_STRIDE` overlaps windows by whole sentences (lifting the prior `PROPOSITION_WINDOW_SENTENCES` ceiling of `le=4`) and `VECTOR_STORE_PROPOSITION_WINDOW_OVERLAP` is the fractional overlap form for budget-based windows.
`VECTOR_STORE_PROPOSITION_ABBREVIATION_AWARE` rejoins fragments split inside abbreviations, initials, or citation runs, and `VECTOR_STORE_PROPOSITION_MEASUREMENT_AWARE` prevents breaks between a number and its unit. `VECTOR_STORE_QUERY_UNIT_SIGNALS_ENABLED` (on by default) adds a query-time lane matching tokens after a number against catalog surface forms without a reindex. `VECTOR_STORE_BM25_TOP_K` sets sparse-lane depth independently of `TOP_K`, and `VECTOR_STORE_FUSION_RANK_CONSTANT` smooths hybrid fusion as weight / (constant + rank). New telemetry fields `queries_truncated` and `query_sequence_limit` are recorded in retrieval metrics.

Tune hybrid retrieval for denser recall by enabling overlapping proposition windows, a higher BM25 lane depth, and two critic passes.

```
$ VECTOR_STORE_PROPOSITION_WINDOW_MAX_TOKENS=512
VECTOR_STORE_PROPOSITION_WINDOW_STRIDE=2
VECTOR_STORE_BM25_TOP_K=20 VECTOR_STORE_FUSION_RANK_CONSTANT=60
FACTS_CRITIC_PASSES=2 ONTOLOGY_CRITIC_PASSES=1 ontocast process
corpus/
```

— Every windowing and fusion knob named with defaults and effect v0.6.4

06

Seed directory flags and zero-seed ontology bootstrapping

NEW

88

`--ontology-dir` and `--shapes-dir` (also `FACTS_SHAPES_DIR`) flags, now on both `serve` and `process`, set or clear seed ontology/shapes directories; passing `--ontology-dir ''` disables seed ontologies so each unit's context resolves empty and the renderer mints a new ontology from the text under `current_domain`. `FACTS_CONTEXT_FROM_UNITS` (on by default) seeds document-level context from unit snapshots in `RENDER_MODE=facts` runs so the aggregator's merge guards and SHACL gate run against a populated vocabulary when no ontology stage runs.

Bootstrap a brand-new ontology from raw documents when you have no seed catalog — let OntoCast mint the schema from scratch, then fold the good output back in as your seed for the next run.

```
$ ontocast ./docs --ontology-dir '' --ontology-output-dir
./ontology-out
```

Bootstrap a run with seed ontologies and shapes so the renderer starts from an established schema rather than building one from scratch.

```
$ ontocast --ontology-dir ./ontologies --shapes-dir ./shapes
<target>
```

Run extraction against a corpus using your own Turtle seed ontologies, so the LLM stays within your domain vocabulary from the start.

```
$ ontocast --ontology-dir ./ontologies/my-domain
```

— Flags, defaults, and bootstrap mechanism named with runnable examples v0.6.4

07 LLM request pacing, retry limits, and failure handling NEW

88

`LLM_REQUESTS_PER_SECOND` (`unset=unpaced`) paces LLM request starts via a per-process token bucket to avoid provider rate-limit errors on tiers such as 500 RPM (~8 RPS). `LLM_MAX_RETRIES` sets the provider SDK transport-retry budget (defaults: OpenAI 2, Anthropic 2, Google 6). New counters `llm/calls_failed` , `llm/timeouts` , `llm/rate_limited` , and `llm/calls_rejected` surface timed-out calls (still charged prompt characters) and provider-rejected calls that were previously hidden. A provider-rejected request (bad key, unsupported parameter, unreachable model) now aborts the batch and exits non-zero naming the provider, model, and rejected parameter, writing no output dumps.

Stay within a 500 RPM provider tier while running many parallel workers, and cap SDK retries to handle residual throttling without amplifying request rate.

```
$ LLM_REQUESTS_PER_SECOND=7 LLM_MAX_RETRIES=4 PARALLEL_WORKERS=8
ontocast run <pipeline>
```

— Token-bucket mechanism, exact defaults, and abort behavior specified v0.6.4

08 Ontology context requirement and domain adherence checks NEW

83

`ONTOLOGY_CONTEXT_REQUIRED` (off by default) raises `EmptyOntologyContextError` per unit when a facts-only deployment receives an empty ontology context, producing a zero-triple manifest and clean exit instead of a silent bad run; `retrieval_metrics.empty_snapshot_reason` records which subsystem caused the empty context. `DOMAIN_ADHERENCE` finding and `FACTS_DOMAIN_ADHERENCE_MIN_SHARE` (default `0.15`) flag renders that received a catalog context but used almost none of it.

Stop a facts-only deployment from silently producing a clean-looking zero-triple run when the curated catalog fails to load.

```ini
ONTOLOGY_CONTEXT_REQUIRED=true
VECTOR_STORE_BM25_TOP_K=20
TOP_K=10
```

— Exact env vars, defaults, and error type named with example v0.6.4

09 Module closure triple budget for ontology chapters NEW

82

`ONTOLOGY_PATCH_SMALL_MODULE_CLOSURE_MAX_TOTAL_TRIPLES` caps total triples pulled from whole-module closures. The run manifest surfaces `module_closure_iris`, `module_closure_declined_iris`, `module_closure_triples`, and `relevance_by_ontology` to report which modules were admitted or excluded by the budget.

Cap total module-closure triples to control chapter size on wide ontologies, then inspect the manifest to see which modules were admitted or crowded out.

```
$ ONTOLOGY_PATCH_SMALL_MODULE_CLOSURE_MAX_TOTAL_TRIPLES=500
ontocast run <pipeline>
```

— Exact env var, runnable example, and manifest fields named `v0.6.4`

---

10 **Ontology chapter format and text-budget controls** NEW

78

`ONTOLOGY_CHAPTER_FORMAT` (`auto|inherit|turtle|term_sheet`) controls the ontology chapter encoding in facts prompts — `term_sheet` replaces serialized RDF with a compact one-line-per-term listing (name, surface forms, hierarchy, connections, scope note) and `turtle` encodes it in Turtle instead of JSON-LD, each reducing prompt tokens without changing extraction semantics; `LLM_GRAPH_FORMAT=turtle` independently encodes the output wire format in Turtle.
`ONTOLOGY_TEXT_MAX_CHARS_NAMING`, `ONTOLOGY_TEXT_MAX_CHARS_CONTRACT`, `ONTOLOGY_TEXT_MAX_CHARS_PROSE`, and `ONTOLOGY_TEXT_TOTAL_BUDGET` cap chapter text by field type and in total. The manifest records `chapter/text_chars_before`, `chapter/text_chars_after`, `chapter/literals_clipped`, and `chapter/text_over_budget` once per memoised snapshot.

— Every format value and budget variable named, no example given `v0.6.4`

---

11 **Unit-scoped fact IRIs for aggregation** NEW

77

`AGG_UNIT_SCOPED_FACT_IRIS` (default `false`) rewrites each unit graph's fact-namespace instance IRIs to `<local>__u<unit index>` after sanitization, so same-name entities minted by independent units form an explicit cluster that the veto gate can dissolve instead of silently fusing into one node.

— Exact env var, default, and IRI format named `product docs`

---

12 **SHACL shapes prompt contract and conformance reporting** NEW

76

`FACTS_SHAPES_PROMPT_CONTRACT` (default `auto`; values `full|context|auto|off`) injects SHACL shapes as `# CONFORMANCE REQUIREMENTS` into the facts prompt, with `FACTS_SHAPES_PROMPT_MAX_LINES`

(default 60) capping conformance chapter size and switching from full-catalog to per-unit context-join selection when exceeded. The SHACL conformance summary gains `shacl_focus_nodes` and `shacl_vacuous` fields, and `conforms` is now `null` when no focus nodes matched instead of `true`.

Enable SHACL-contract enforcement and numeric-coverage checking to catch under-specified facts before they enter the graph.

```
$ FACTS_SHAPES_PROMPT_CONTRACT=full
FACTS_SHAPES_PROMPT_MAX_LINES=200
FACTS_NUMERIC_COVERAGE_MANDATORY=true
FACTS_NUMERIC_COVERAGE_LIMIT=50 ontocast process corpus/
```

— Exact values, defaults, and behavior change named `v0.6.4`

13 **CLI flag for batch facts user instructions** NEW 69

`--facts-user-instruction` on `ontocast process` lets batch runs carry the same deployment guidance as server requests without exposing the instruction text in run manifests — only its length is recorded as `validation_config.facts_user_instruction_chars`.

Process a directory of documents with sensitive deployment guidance kept out of the shareable run manifest.

```
$ ontocast process --input-path ./docs --facts-user-instruction -
-output-dir ./out
```

— Exact flag and manifest field named with a runnable example `v0.6.4`

14 **Retrieval recall test harness** NEW 69

Ships `test/test_retrieval_recall.py` and `test/retrieval_gt.py` to report the retrieval funnel against a user-supplied catalog, pointed at a corpus via `ONTOCAST_RECALL_CORPUS`, with `ONTOCAST_RECALL_COLLECTION_SUFFIX`, `ONTOCAST_RECALL_SKIP_INDEX`, and `ONTOCAST_RECALL_JSON` for seed vs. snapshot recall testing without an LLM.

— Named test files and env vars but no usage example given `v0.6.4`

15 **Reasoning effort and thinking-budget controls** NEW 67

`LLM_REASONING_EFFORT` (OpenAI reasoning models and Gemini 3+; values `none|minimal|low|medium|high|xhigh`) and `LLM_THINKING_BUDGET` (Gemini 2.5 integer spelling: `0`=off, `-1`=model-chosen, `N`=cap, superseded by

`LLM_REASONING_EFFORT` from Gemini 3 on) bound reasoning-model output cost when thinking tokens dominate.

— Exact values and provider scoping named, no example `v0.6.4`

16 **Constrained JSON decoding for OpenAI** NEW 66

`LLM_JSON_MODE=true` (off by default, OpenAI only) constrains decoding to syntactically valid JSON via `response_format: json_object`, making JSON syntax errors unreachable rather than repaired after the fact; recorded as `validation_config.json_mode` in the run manifest.

— Exact env var and mechanism named, no runnable example `product docs`

17 **Chunk-level provenance retention flag** NEW 65

`--keep-provenance` retains chunk-level provenance in the facts dump, enabling source-span tracing, re-verification, and auditing of extracted facts.

Retain chunk-level provenance in the facts dump so extracted facts can be traced back to their source spans for auditing or re-verification.

```
$ ontocast --keep-provenance
```

— Exact flag and runnable example, but thin on mechanism `product docs`

18 **Numeric export-artifact repair** NEW 64

`CONVERTER_REPAIR_NUMERIC_ARTIFACTS` (off by default, not part of the `born_digital` path) repairs pattern-local export artifacts inside values, including HTML entities, carriage-return column wraps, and flattened exponents.

— Exact env var and artifact types named, no example `product docs`

19 **Self-describing run manifest fields** NEW 64

The manifest's `validation_config` block records `context_from_units`, `json_mode`, `shapes_prompt_contract`, `shapes_triples`, `shacl_inference`, `numeric_coverage_mandatory`, and `facts_user_instruction_chars` so each output directory is self-describing for sweeps. The `prompting` section additionally gains `prompt_cache_key`, `max_inflight`, `llm_graph_format`, `ontology_chapter_format`, `ontology_context_scope`, `fanout_warmup_units`, `parallel_workers`, and `embedding_model_name` fields.

— Every recorded field named, but purely descriptive `v0.6.4`

20 **Startup configuration validation checks** NEW 62

Rejects startup when `ONTOLOGY_PATCH_MMR_LAMBDA < 1.0` is combined with non-zero atom floors, an incompatible configuration caught before any work runs. Warns at

startup when `ONTOLOGY_PATCH_MAX_ATOMS` exceeds the window-scaled ceiling, naming the effective cap. Adds a catalog/index agreement check at startup: an empty catalog paired with a populated index now fails, and extra indexed IRIs warn.

— Exact conditions named, no direct usage example given v0.6.4

THINNER COVERAGE BELOW

21 **Removed HYBRID cross-query merge mode** BREAKING 44

The `CrossQueryMergeMode.HYBRID` enum value is removed; configs or code referencing `HYBRID` will break.

— Names the removed enum but gives no migration guidance v0.6.4

└──▷ **BREAKING ON UPGRADE**

! The `CrossQueryMergeMode.HYBRID` enum value is removed; configs or code referencing `HYBRID` will break.

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling added NVIDIA Nemotron Parse 2.0 as both a model-catalog entry and a VLM pipeline option, introduced a rich Chandra HTML output format that preserves tables, forms and picture metadata, and added minimal AFP document support alongside a refactored PDF backend that drops the pypdfium dependency.

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

01 **Chandra HTML output format** NEW 96

Adds `ResponseFormat.CHANDRA_HTML` with `CHANDRA_OCR_LAYOUT_PROMPT` to parse Chandra model output as HTML layout blocks, scaling `data-bbox` coordinates from 0–1000 to source page size. Preserves paragraphs, heading levels, nested lists, table spans, inline formatting, links, math, and code whitespace, including tables found inside blocks labeled `Text`, `Form`, or `Figure`. Structured table cells are stored via `RichTableCell` references, and form region checkbox, radio, and fillable text values are retained without inferring key–value links from visual proximity. Picture `img alt` descriptions are stored in `PictureItem.meta.description` and explicit `<chem>` content as SMILES molecule metadata; enabling `generate_picture_images` or `generate_page_images` lets picture.get_image(document) crop pictures using provenance coordinates, and `traverse_pictures=True` includes picture children when exporting Markdown. Returns `PARTIAL_SUCCESS` (rather than a silent empty result) when nonempty prose or JSON responses contain no HTML transcription, and emits warnings for unrecognized layout or invalid bounding boxes.

Export a Chandra-parsed document to Markdown and include all picture children (chart tables, diagram code, etc.).

```python
md = doc.export_to_markdown(traverse_pictures=True)
print(md)
```

— Many named APIs with a runnable export example and behavior detail product docs

02 **NVIDIA Nemotron Parse 2.0 model support** NEW 75

Adds `nemotron_parse_v2` (NVIDIA Nemotron Parse 2.0, 0.9B) to the model catalog, supporting native layout parsing with Markdown text and LaTeX table output, and exposes it as a supported VLM pipeline model via `nvidia/NVIDIA-Nemotron-Parse-2.0`, usable from the CLI with `--pipeline vlm --vlm-model`.

Use NVIDIA-Nemotron-Parse-2.0 as the VLM for high-quality visual document understanding via the CLI.

```
$ docling --pipeline vlm --vlm-model nvidia/NVIDIA-Nemotron-Parse-2.0 https://arxiv.org/pdf/2206.01062
```

— Named model ID and runnable CLI command given `v2.128.0`

THINNER COVERAGE BELOW

03 **docling-parse backend refactor** IMPROVED 40

Refactors the `docling-parse` PDF backend to remove the `pypdfium` dependency.

— Names dependency removed but no migration guidance given `v2.128.0`

04 **AFP document format support** NEW 30

Adds minimal AFP (Advanced Function Presentation) document format support to the parsing pipeline.

— No usage example or flags shown for AFP parsing `v2.128.0`

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ 1 RELEASE · 2026-08-20

NOTES ↗ RANK

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

> LangSmith tightened annotation queue batching, deprecated the legacy feedback formula API in favor of composite evaluators, and removed legacy dataset comparison helpers from its public SDKs.

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

01 200-item limit on annotation queue additions IMPROVED

80

`POST /annotation-queues/<id>/items` now accepts at most 200 items per request and returns a clear validation error when the limit is exceeded. A single request can mix individual runs and conversation threads (using `item_type: THREAD` with `thread_id` and `session_id`) for combined human review.

Add a batch of runs and a conversation thread together to an annotation queue for mixed human review.

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

— Names endpoint, exact limit, and includes a runnable curl example. snapshot-20260917

└──▷ BREAKING ON UPGRADE

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

! Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; use `POST /v2/datasets/<id>/experiment-runs` instead (existing HTTP routes still work for LangSmith UI clients).

WAS THIS USEFUL? 👍 👎

MLflow

Feedback ⓘ 1 RELEASE · 2026-09-17 NOTES ↗ RANK

MLflow tracks, evaluates, and monitors machine learning and LLM applications, covering experiment tracking, tracing, prompt management, and a model registry.

> MLflow 3.16.1 adds scorer execution timeouts, scorer version tracking in evaluation metadata, and cross-trace span links for Unity Catalog, while removing the default basic-auth admin password for security.

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

01 Default basic-auth admin password removed BREAKING

70

The default basic-auth admin password has been removed from `basic_auth.ini`; deployments relying on the shipped default password will lose admin access on upgrade and must set an explicit password.

– Names exact config file and required migration action v3.16.1

02 Timeout option for evaluation scorers NEW

60

The `@scorer` decorator now supports a `timeout` option to bound how long an evaluation scorer can run, preventing runaway scorer execution during evaluation.

– Names exact decorator and param but no default or example v3.16.1

THINNER COVERAGE BELOW

03 Span links for Unity Catalog traces NEW

45

MLflow now supports span links for Unity Catalog traces, enabling cross-trace relationships within Databricks Unity Catalog.

– Names the surface but not how to create or view links v3.16.1

04 Scorer version in assessment metadata IMPROVED

40

Scorer version is now propagated to assessment metadata, making scorer versioning visible directly in evaluation results.

– States what changed but not the mechanism or metadata field name v3.16.1

└──▷ **BREAKING ON UPGRADE**

! The default basic-auth admin password has been removed from `basic_auth.ini` — deployments relying on the shipped default password will lose admin access on upgrade and must set an explicit password.

WAS THIS USEFUL?  

agentacct

Feedback ⓘ 1 RELEASE · 2026-09-17 NOTES ↗ RANK

The agentacct dashboard records coding-agent work steps, tools, file changes, tests, time, and token costs locally.

> agentacct's biggest addition this window is Work Receipts — local per-task evidence documents with Verified/Reported/In Progress/Observed verdicts — alongside a new cross-agent Work timeline, DeepSeek Harness support, and a batch of onboarding, TUI and capability-check CLI commands.

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

01 Onboarding, install-scope and lifecycle commands NEW

81

Adds `agentacct onboard` to detect installed coding agents, set up a local store, and start the session recorder in one step; a `--scope project` install option to keep all state inside a repo's `.agent-sentinel/` directory instead of the global store; `agentacct setup prompt --agent` to generate an install prompt a coding agent can follow to self-install; `agentacct uninstall-autostart` to remove the autostart entry created by `agentacct onboard`; and `agentacct stop` to stop the managed sync process and local API (owned processes only).

First-time setup on a new machine: discover installed agents, initialize the local store, and start recording sessions.

```
$ agentacct onboard
```

Bootstrap agentacct on a new machine — detects all running coding agents, wires up hooks and MCP servers, and starts the local recorder.

```
$ agentacct onboard
```

— Five exact commands and flags, one runnable in the example product docs

02 Per-agent coverage matrix command NEW

76

Adds `agentacct capabilities agents` command to print the per-agent coverage matrix (Claude Code, Codex, OpenCode, Hermes, DeepSeek Harness, OpenClaw, Cursor) for the installed version.

Check which agent integrations are fully supported before trusting a receipt — prints the coverage matrix for your installed version.

```
$ agentacct capabilities agents
```

Check which evidence lanes (receipt detail, cost join, step recording) are actually available for each agent you have installed before trusting a run's verdict.


```
$ agentacct capabilities agents
```


— Exact command and full agent list with a runnable example `product docs`

---

03 **DeepSeek Harness agent support** NEW 75

Adds DeepSeek Harness (`dsh`) as a supported agent with local usage import from its compressed session logs, brought in via `agentacct onboard --agent dsh` for one-command MCP self-reporting onboarding.

Onboard DeepSeek Harness so agentacct can self-report MCP usage and import its local session logs.

```
$ agentacct onboard --agent dsh
```

— Exact onboarding command and import mechanism named `v0.11.0`

---

04 **Local-only recording and storage model** NEW 75

Records tool names, touched files, credential-scrubbed commands, exit codes, tokens, and steps — explicitly excluding prompts, model responses, and transcripts — per the published privacy threat model. All state is stored locally at `~/.local/state/agentacct/state` ; the app listens only on `127.0.0.1` and never contacts provider APIs or external services.

— Exact storage path and bind address given, no user action needed `product docs`

---

05 **Terminal UI reaches feature parity with macOS app** IMPROVED 75

Adds `agentacct tui` to view the Shift Brief, Work Receipts, capacity, and cross-agent session timelines from the terminal, later brought to five-tab parity with the macOS app including the new Work timeline with keyboard open, zoom, and scrub.

— Exact command and named tabs, interaction detail is thin `v0.11.0`

---

06 **Cross-agent Work timeline and Sessions tab** NEW 71

Adds a cross-agent Work timeline that groups a folder's sessions across Claude Code, Codex, OpenCode, and DeepSeek Harness onto a single timeline, available in both the

macOS app and `agentacct tui`. The former Work tab was renamed to Sessions, with the new Work tab now showing this cross-agent folder timeline.

— Names the agents and UI surfaces but only a navigation path v0.11.0

07 **Work Receipts with per-step evidence tiers** NEW 69

Introduces Work Receipts — per-task evidence documents combining agent-reported steps, hook-observed exit codes, and CI checks into a single Verified/Reported/In Progress/Observed verdict. One receipt is generated per task from local session logs, classifying each step and assigning each check an evidence tier (agent-reported, hook-observed exit code, or CI), all processed locally with nothing leaving the machine.

— Clear mechanism and verdict states but no command or UI path given product docs

08 **Per-client provider quota tracking** NEW 65

Tracks per-client provider quota windows (e.g. 5-hour and weekly windows for Codex and Claude Code) alongside each agent's seven-day recorded usage, labeled as pricing estimates (`≈`).

— Concrete windows and labeling named, viewed only via dashboard product docs

09 **Dashboard Shift Brief** NEW 65

Introduces a Dashboard Shift Brief that surfaces the one task most needing human attention, its recorded reason, and its provenance (e.g. MCP record), with a Copy review brief action that copies only recorded facts.

— Named UI action and provenance concept, no command given product docs

10 **Confidence-scored usage-to-work attribution** NEW 60

Joins usage and work records by real session and transcript IDs, labeling every attribution with a confidence level (`exact`, `high`, `medium`, `low`) and showing unproven links as gaps rather than zeroes.

— Confidence levels named but no way to inspect them stated product docs

THINNER COVERAGE BELOW

11 **Session and Work detail records** NEW 57

Adds readable Session and Work detail records with Steps/Checks outcome bars, a numbered step spine with checks and files inline, and an activity timeline.

— Names UI elements but no command or exact path v0.11.0

12 **Native macOS app and Python CLI distribution** NEW 50

Ships a signed, notarized macOS app (macOS 14+) for a native window experience without requiring Python, alongside the Python CLI (Python >= 3.11, macOS or Linux; Windows via WSL).

— Platform and version requirements named, no setup steps given `product docs`

WAS THIS USEFUL?  

Arize Phoenix

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

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

> Phoenix 20.13.0 makes datasets editable in the UI and via GraphQL, replaces the `rootSpansOnly` GraphQL field with a new span filter DSL, and adds SQLite time-function support for latency queries.

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

01 **Span filter DSL replaces rootSpansOnly** BREAKING 65

The `rootSpansOnly` field has been removed from the GraphQL API; span filtering must now use the new span filter DSL, which is documented in the skills.

– Names removed field and replacement but no DSL syntax examples `arize-phoenix-v20.13.0`

02 **sqlean time extension for SQLite backends** NEW 60

Adds `sqlean` time extension support for `date_trunc` and `latency_ms` on SQLite backends.

– Names exact functions and backend but no usage example `arize-phoenix-v20.13.0`

THINNER COVERAGE BELOW

03 **Editable dataset examples table** NEW 40

The UI now includes an editable dataset examples table, allowing inline editing of dataset examples directly in the interface.

– Names the UI feature but no navigation path or field detail `arize-phoenix-v20.13.0`

04 **JSON Patch-style dataset mutations via GraphQL** NEW 30

Dataset mutations can now be performed via GraphQL using a JSON Patch-style approach.

– Mentioned only in summary with no mechanism or schema detail `arize-phoenix-v20.13.0`

└──▷ **BREAKING ON UPGRADE**

! The `rootSpansOnly` field has been removed from the GraphQL API; span filtering must now use the span filter DSL.

WAS THIS USEFUL? 👍 👎

ai-gateway

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ RANK

AI Gateway routes AI model requests through a self-hosted gateway with observability and security controls.

> ai-gateway v1.5.8 adds programmatic PII and prompt-injection detector functions, an observe-only PII logging mode, and expands secret-scan coverage with seven new kinds — while reclassifying Anthropic key detection in a breaking change.

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

01 **PII and prompt-injection detector functions** NEW 75

Adds `piiredact.Detect` and `promptshield.Detect` pure functions that report, by name, which built-in PII entity types or injection categories occur in a string — without returning matched text — so an embedding policy layer can query detection results directly.

Query PII detection results programmatically in a policy layer without exposing matched text.

go

```go
hits := piiredact.Detect(inputText)
for _, entity := range hits {
    log.Printf("PII entity detected: %s", entity)
}
```

— Names two functions with a runnable code example v1.5.8

02 **New secret-scan kinds for GitLab, npm and more** NEW 70

Adds seven new `secret-scan` kinds matched on structure alone: `gitlab_token`, `npm_token`, `huggingface_token`, `sendgrid_key`, `twilio_key`, `azure_storage_key`, and `slack_webhook`.

Extend secret scanning to cover GitLab tokens and Slack webhooks in addition to existing kinds.

```yaml
plugins:
  - name: secret-scan
    config:
      kinds:
        - gitlab_token
        - slack_webhook
        - anthropic_key
```

— All seven kinds named but detection logic unexplained v1.5.8

---

03 **Observe-only mode for PII redaction** NEW 65

Adds `action: log` to `pii-redact`, enabling an observe-only mode that detects and records PII without denying or rewriting requests.

Enable observe-only PII detection on a route to audit traffic before deciding whether to enforce redaction.

```yaml
plugins:
  - name: pii-redact
    config:
      action: log
```

— Exact config key and working YAML snippet given v1.5.8

---

04 **Anthropic keys reclassified in secret-scan** BREAKING 60

Anthropic API keys are now reported under `anthropic_key` in `secret-scan`; any allow/deny rule using `kinds: ["openai_key"]` to catch Anthropic keys will no longer match them.

— States before/after but no migration snippet v1.5.8

└──▷ **BREAKING ON UPGRADE**

! Anthropic API keys are now reported under `anthropic_key` in `secret-scan`; any allow/deny rule using `kinds: ["openai_key"]` to catch Anthropic keys will no longer match them.

WAS THIS USEFUL? 👍 👎

Braintrust

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

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

> Braintrust's biggest addition this window is a Debugger that diagnoses failure modes in individual agent traces, alongside moving Loop onto a managed runtime for longer-running, cross-surface investigations; smaller updates cover topic-distribution filtering, GPT-5.6 Luna support, Grok trace metadata, a custom CA bundle option, and removal of a deprecated topics flag.

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

01 Custom CA bundle for bt tracing NEW

75

Supports trusting a corporate CA when running `bt trace` in environments with a private TLS certificate chain, via the `BRAINTRUST_CUSTOM_CA_BUNDLE` environment variable, e.g. `export BRAINTRUST_CUSTOM_CA_BUNDLE="$(cat /etc/ssl/certs/corp-ca.pem)"` followed by `bt trace run pi`.

Trust a corporate CA when running bt tracing in an environment with a private TLS certificate chain.

```
$ export BRAINTRUST_CUSTOM_CA_BUNDLE="$(cat /etc/ssl/certs/corp-ca.pem)"
bt trace run pi
```

— Names exact env var and gives a runnable command snapshot-20260917

02 Debugger for agent trace failure modes NEW

65

Debugger reports likely failure modes for a single agent trace, citing relevant spans, tool calls, and model outputs, and ships with an Analyze trace view that groups a run into labeled Work sections. Requires data plane v2.12.0+.

— Explains mechanism and prerequisite version, but no UI path or command given

snapshot-20260917

03 Loop moved to Braintrust-managed runtime IMPROVED

65

Loop now runs on a Braintrust-managed runtime so threads persist and investigations can run longer; Loop can operate across logs, experiments, and datasets, and can create or edit prompts, scorers, datasets, facets, custom views, dashboards, and automations. Requires data plane v2.9.0+.

— Describes scope and requirements but no concrete usage step snapshot-20260917

THINNER COVERAGE BELOW

04 **Search bar filtering for live topic distributions** NEW

40

Adds a filter input via the search bar when viewing live topic distributions, with autocomplete to help build filters.

— Clear UI entry point but no further mechanism detail `product docs`

---

05 **Grok metadata in trace spans** NEW

35

Adds Grok repository metadata and local username to Grok session spans in trace metadata.

— Names the fields added but no usage path `snapshot-20260917`

---

06 **GPT-5.6 Luna max reasoning effort support** NEW

10

Braintrust adds support for GPT-5.6 Luna max reasoning effort.

— Only named in summary, no detail on usage or scope `product docs`

└──▷ **BREAKING ON UPGRADE**

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set` in bt v0.20.0; the flag was deprecated and ignored since data plane v2.15.0.

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 1 RELEASE · 2026-09-17 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik 2.2.67 focuses on backend groundwork for automated annotation queues and improves MCP-related identity capture and debugging context in the UI.

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

01 **MCP OAuth client metadata capture** NEW 40

Opik now captures RFC 7591 client software metadata during MCP OAuth registration, improving identity context for MCP-connected clients.

— Names RFC standard but no config/endpoint detail 2.2.67

02 **MCP hint on trace error UI** IMPROVED 40

Expanded trace errors in the UI now surface an MCP hint, giving practitioners a contextual starting point when debugging failed traces.

— Names UI location but no exact navigation steps 2.2.67

03 **Annotation queue automation data model** NEW 30

Adds a new annotation queue automation data model and backend services, laying the foundation for automated annotation queue workflows.

— Foundational backend change, no user-facing surface yet 2.2.67

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

AI-Infra-Guard

Feedback ⓘ 1 RELEASE · 2026-09-17 NOTES ↗

RANK

AI-Infra-Guard is an AI red teaming platform that scans AI infrastructure for CVE vulnerabilities, MCP/Agent skill risks, jailbreak weaknesses, and API relay security issues.

> AI-Infra-Guard's v4.6.2 release focuses on expanding its Research benchmarking suite for autonomous agent red teaming and refreshing its detection rule set.

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

01 **FORGE-Bench and RogueHandoff-20 agent red-teaming benchmarks**

NEW 55

Adds `forge_bench` (FORGE-Bench), a deterministic Loss-of-Control benchmark for evaluating autonomous agents, shipped with a built-in OpenAI-compatible client and a bilingual README. Also adds the RogueHandoff-20 benchmark under the Research section for evaluating rogue handoff scenarios in autonomous agents.

— Names both benchmarks and client but no run instructions or scoring details v4.6.2

02 **AIG detection rules data refresh** IMPROVED 25

Updates the AIG rules data set, dated 2026-09-11.

— Only a date is given, no scope of rule changes v4.6.2

WAS THIS USEFUL? 👍 👎

NoLabs nono

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗ CODE ↗

RANK

The nono runtime runs AI agents in isolated environments with policy-controlled filesystem, network, and credential access.

> nono v0.78.0 adds backend-signed approval webhooks, a macOS API to block implicit DNS grants, and a prebuilt Nix flake output for faster installs.

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

01 Backend-signed approval webhooks via platform auth NEW

88

A new `auth` field on `ApprovalBackendConfig` (value `platform`) signs every approval webhook request with the client's platform enrollment key (`nono-request-v1`), defaults `url` to the enrolled platform's approval route, and switches the backend into submit-and-poll mode. Configured via the `approval_backends` config block with `backend_type: webhook` and `auth: platform`.

Configure a webhook approval backend that authenticates via the platform enrollment key and uses submit-and-poll, so an external policy service can sign off on agent actions.

json

```
{
  "approval_backends": {
    "my-platform-backend": {
      "backend_type": "webhook",
      "auth": "platform",
      "timeout_secs": 30
    }
  }
}
```

— Names exact config field, values, and signing key; full config example given v0.78.0

02 macOS DNS opt-out for network capability sets NEW

73

Adds CapabilitySet::block_dns() and the C API nono_capability_set_block_dns() / nono_capability_set_dns_enabled() on macOS to opt out of the implicit `mDNSResponder` grants that `Blocked` and `ProxyOnly` network modes otherwise apply, letting Nix derivations and other callers prevent host DNS queries entirely.

— Names exact API functions and modes but has no runnable example v0.78.0

03 **Prebuilt Nix flake output** NEW 72

Adds a `#prebuilt` Nix flake output, runnable as `nix run github:nolabs-ai/nono#prebuilt`, that fetches the official release binary directly from GitHub Releases instead of compiling from source.

Pin and run the prebuilt nono binary in a Nix environment without triggering a full Rust compilation.

```
$ nix run github:nolabs-ai/nono#prebuilt
```

— Exact runnable command given for the new flake output v0.78.0

WAS THIS USEFUL? 👍 👎

Mend.io Mend AI

Feedback ⓘ 1 RELEASE · 2026-09-16 NOTES ↗

RANK

Mend AI is an AI security service that discovers AI components and helps teams govern their use in software applications.

> Mend AI's window centered on a breaking authentication change to the Guardrails SDK, alongside several Renovate configuration and dependency-handling improvements.

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

01 Guardrails SDK requires Mend Activation Key BREAKING

65

Guardrails SDK downloads now require the Mend Activation Key as an authentication step, in addition to the existing authentication process used during SDK usage. Existing Guardrails SDK installations with unpinned versions that attempt to re-install dependencies will now be required to provide the Mend Activation Key during the download procedure.

– Names exact mechanism and migration impact, no code example given (16-September-2026)

02 `minimumReleaseAgeBuffer` config option in Renovate NEW

65

Adds `minimumReleaseAgeBuffer` option to Renovate config, giving practitioners a buffer on top of the existing minimum release age setting.

– Names exact config key but not its values or defaults product docs

03 Alpine apk add package extraction from Dockerfiles NEW

65

Renovate now extracts `apk add` packages from `RUN` instructions in Dockerfiles, enabling it to manage Alpine package versions declared inline in Dockerfile `RUN` layers.

– Explains scope and mechanism but no config flag to enable it product docs

THINNER COVERAGE BELOW

04 Direct GitHub merge before queue fallback IMPROVED

45

Renovate now attempts a direct merge on GitHub before falling back to the merge queue, reducing unnecessary merge-queue usage when a direct merge is possible.

– Describes behaviour change but no configuration surface named product docs

└──▷ **BREAKING ON UPGRADE**

! Existing Guardrails SDK installations with unpinned versions that try to re-install dependencies will now require providing the Mend Activation Key during the download procedure.

WAS THIS USEFUL?  

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 17, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue