THE AI TOOLCHAIN

NO. 039

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 27, 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 94 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 24 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. Anthropic's on-demand context compaction lets a long agent run shed its own history mid-conversation ad of dying at the window; SGLang's Encoder-Prefill-Decode

vim ? help

disaggregation scales the three phases of a multimodal request on separate hardware; and Pipelock now scans MCP tool descriptions and arguments before a call leaves the machine.

DEPLOY

Keep a long agent session alive past the context window without rebuilding your own summarizer

Anthropic added on-demand context compaction and inline MCP tool definitions, so tools can be declared per-call rather than pre-registered; Strands' Harness SDK gained context management with prompt caching controls and on-demand Agent Skills loading, and KoboldCpp brought compaction to its local Agent. Teams currently hand-roll trimming and summarization logic and discover its bugs when a run silently forgets its instructions.

Anthropic · Strands Agents · KoboldCpp

DEPLOY

Scale the prefill, decode and encoder stages of a serving cluster independently

SGLang shipped Expert Parallelism, Decode Context Parallelism and Encoder-Prefill-Decode disaggregation, plus quantized FP8/FP4 KV cache and a weight-free scheduling simulator you can plan a topology against before buying the GPUs. LLM Compressor's NVFP4 example covers FP8-to-BF16 conversion and selective quantization of MoE expert weights, so the experts that hurt accuracy can be left alone.

SGLang · LLM Compressor

GOVERN

Block a prompt-injected MCP tool before the call reaches the server

Pipelock now scans tool descriptions (the vector where a malicious server hides instructions), applies DLP to arguments, sandboxes stdio servers on Linux, and signs mediation receipts with RFC 9421 so you have audit evidence of what was allowed. Foremerge closed a command-injection path by restricting changeset validation to a trusted registry of checks.

Pipelock

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.106 NOTES ↗ RANK

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

> xalgorix's agent gained blind-XXE detection, configurable business-logic scan depth, and wordlist gating in this release.

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

01 **Blind-XXE verification in agent** NEW 30

The agent now performs blind-XXE verification, enabling out-of-band XML External Entity detection.

— Names the technique but no config or command detail v4.6.106

02 **Business-logic lane depth configuration** NEW 28

The agent adds business-logic lane depth configuration, allowing control over scan coverage depth.

— Describes purpose but no specific flag or values given v4.6.106

03 **Wordlist gate for scanning** NEW 28

A wordlist gate was added to the agent to control which wordlists are used during scanning.

— States function but no configuration surface named v4.6.106

v4.6.106 NOTES ↗ LATEST STABLE

xalgorix v4.6.106 adds blind-XXE verification, business-logic lane depth control, and a wordlist gate to the agent.

└──▷ GET THIS VERSION

```
$ git clone --branch v4.6.106
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.106
```

Adds blind-XXE verification to the agent, enabling out-of-band XML External Entity detection.

> Adds business-logic lane depth configuration to the agent for controlling scan coverage depth.
>
> Adds a wordlist gate to the agent to control which wordlists are used during scanning.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Replit Agent

Feedback ⓘ snapshot-20260927 NOTES ↗ RANK

Replit Agent is an AI coding agent that builds, tests, and deploys applications in Replit.

> Replit Agent shipped a Muse integration that turns in-conversation ideas into runnable Replit apps, added a Workspace access-requests workflow for usage-limit and publishing approvals, and exposed a `/llms.txt` documentation index endpoint for programmatic discovery.

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

01 **/llms.txt documentation index endpoint** NEW 70

Adds a `/llms.txt` endpoint that returns the complete documentation index, enabling discovery of all available pages before further exploration; it can be fetched directly with `curl https://docs.replit.com/llms.txt`.

Discover all available Replit Agent documentation pages before scripting or automating further doc fetches.

```
$ curl https://docs.replit.com/llms.txt
```

— Named endpoint with a runnable curl command snapshot-20260927

snapshot-20260927 NOTES ↗ STABLE

Replit Agent exposes a documentation index at `/llms.txt` for programmatic discovery of all available pages.

└──▷ **TRY IT**

Discover all available Replit Agent documentation pages before scripting or automating further doc fetches.

```
$ curl https://docs.replit.com/llms.txt
```

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-08-28 CODE ↗

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

Adds waterfall chart, heatmap, Streamgraph, Marimekko, Excalidraw import, and new semantic patterns plus a print-a3-landscape size preset.

└──▷ **GET THIS VERSION**

```
$ git clone --branch commits-2026-08-28
https://github.com/cathrynlavery/diagram-design.git
# already have the repo? check out this version:
$ git checkout commits-2026-08-28
```

Adds `print-a3-landscape` size preset to the output spec for print-ready diagram exports.

Adds a waterfall chart as a new visual type.

Adds heatmap as visual type #40.

Adds Streamgraph as a Line variant.

Adds Marimekko as a Treemap variant, with `scripts/verify-marimekko.py` enforcing nine geometry invariants (column-share widths, segment tiling, area encoding) against declared amounts.

+5 more

WAS THIS USEFUL? 👍 👎

OpenAI Codex CLI

Feedback ⓘ rust-v0.159.0-alpha.7 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 proxy option for routing private IP traffic through an upstream proxy in exec-server, expanded and defaulted its TUI copy-on-select behavior, and added several smaller terminal UI refinements including clipboard shortcuts, Markdown table copy preservation, and a Windows daemon fallback.

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

01 **Configurable copy-on-select in TUI** IMPROVED 86

The `tui.copy_on_select` config key now defaults to `"auto"`, copying selected text on mouse release for most terminals, including unknown terminals and older Ghostty versions, while exempting Ghostty ≥1.2.0, Kitty on macOS, Windows Terminal, and VS Code on Windows. It can also be set explicitly to `always` (force copy on every mouse release) or `never` (disable copy-on-select entirely) via `tui:\n copy_on_select: always` or `never` in config.

Set copy-on-select to always copy on mouse release regardless of terminal, overriding the auto-detection logic.

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

Force copy-on-select on every mouse release regardless of terminal, overriding the new `auto` default.

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

Disable copy-on-select entirely when your terminal handles clipboard natively and you want no interference.

```yaml
tui:
  copy_on_select: never
```

— Config key, default, and override values all named `rust-v0.159.0-alpha.5` `rust-v0.158.0-alpha.15.2`

02 **Upstream proxying for private IPs in exec-server** NEW

85

Adds a `--proxy-private-ips-via-upstream` flag and `CODEX_EXEC_SERVER_PROXY_PRIVATE_IPS_VIA_UPSTREAM` environment variable so permitted private (RFC-1918) IP traffic is routed through the configured upstream proxy in exec-server, useful when network policy requires all outbound connections to traverse a proxy.

Route exec-server traffic destined for internal private IPs through your upstream proxy — useful when your network policy requires all outbound connections, including RFC-1918 addresses, to traverse a proxy.

```
$ codex exec-server --proxy-private-ips-via-upstream
```

— Names flag, env var, and a runnable command `rust-v0.159.0-alpha.7`

THINNER COVERAGE BELOW

03 **Clipboard shortcut in onboarding flow** NEW

50

A `c` shortcut in the TUI onboarding flow copies the browser sign-in URL to the clipboard, with status feedback shown for pending, successful, unconfirmed, busy, and failed clipboard requests.

— Names the key and states but no example `rust-v0.159.0-alpha.7`

04 **Markdown table preservation on copy** IMPROVED

45

Copying TUI responses now preserves Markdown table structure, cell alignment, inline formatting, and whitespace, replacing the previous behavior of collapsing selected tables into code blocks.

— Clear before/after but no invocation detail `rust-v0.159.0-alpha.7`

05 **Embedded server fallback on Windows launchers** IMPROVED

45

When automatic daemon startup is blocked by restrictive Windows launchers such as `cargo run`, Codex now falls back to embedded server mode and shows a visible warning.

— Names the trigger and fallback but no invocation `rust-v0.158.0-alpha.15.2`

06 **Turn tips shown during agent work** NEW

20

The TUI now shows turn tips while an agent is working and after completion.

— Bare description with no mechanism `rust-v0.158.0-alpha.15.2`

---

07 **Refreshed TUI welcome screen** IMPROVED 10

The TUI welcome screen shown for new sessions has been refreshed.

— Only a name, no detail on changes `rust-v0.158.0-alpha.15.2`

`rust-v0.159.0-alpha.7` NOTES ↗ CODE ↗ PRE-RELEASE

Adds `--proxy-private-ips-via-upstream` flag and `CODEX_EXEC_SERVER_PROXY_PRIVATE_IPS_VIA_UPSTREAM` env var to route permitted private IPs through the configured upstream proxy in exec-server.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.159.0-alpha.7
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.159.0-alpha.7
```

└──▷ TRY IT

Route exec-server traffic destined for internal private IPs through your upstream proxy — useful when your network policy requires all outbound connections, including RFC-1918 addresses, to traverse a proxy.

```
$ codex exec-server --proxy-private-ips-via-upstream
```

Adds a `c` shortcut in the TUI onboarding flow to copy the browser sign-in URL to the clipboard, with status feedback for pending, successful, unconfirmed, busy, and failed clipboard requests.

Preserves Markdown table structure, cell alignment, inline formatting, and whitespace when copying TUI responses, replacing the previous behavior of collapsing selected tables into code blocks.

`rust-v0.158.0-alpha.15.2` NOTES ↗ CODE ↗ PRE-RELEASE

OpenAI Codex CLI: Expands `tui.copy_on_select = "auto"` to copy on mouse release for unknown terminals and older Ghostty versions, while exempting Ghostty ≥1.2.0, Kitty on macOS, Windows Terminal, and VS Code on Windows.

└──▷ GET THIS VERSION

```
$ git clone --branch rust-v0.158.0-alpha.15.2
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.158.0-alpha.15.2
```

└──▷ **USE IT**

Set copy-on-select to always copy on mouse release regardless of terminal, overriding the auto-detection logic.

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

Shows turn tips while an agent is working and after completion in the TUI.

Refreshes the TUI welcome screen for new sessions.

Falls back to embedded server mode with a visible warning when automatic daemon startup is blocked by restrictive Windows launchers such as `cargo run`.

WAS THIS USEFUL? 👍 👎

Red Hat ripwire

Feedback ⓘ v0.6.5 NOTES ↗ RANK

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

> ripwire's v0.6.5 release resolves TypeScript path-alias imports as real call-graph edges and adds a `declined_calls=` counter for ambiguous method bindings, alongside new cache warm-up diagnostics via `RIPWIRE_CACHE_STATS` and expanded Windows `--doctor` checks.

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

01 declined_calls= counter for ambiguous method bindings NEW

91

A new `declined_calls=` output field in Python, JS/TS and Ruby counts method calls on a dict or map that cannot be bound to a specific in-repo method because the calling file doesn't name the class (or one in its inheritance cone). Check it with `ripwire . --for="process_payment" 2>&1 | grep declined_calls=`.

Check which calls ripwire declined to bind — useful when auditing a Python or TS codebase where dict/map method calls are widespread and you want to know what coverage is missing.

```
$ ripwire . --for="process_payment" 2>&1 | grep declined_calls=
```

— Names field, three languages, and exact grep command. v0.6.5

02 Windows diagnostics in --doctor IMPROVED

83

`ripwire . --doctor` on Windows reports `binary-path`, `which=`, `which_version=`, `same_bytes=`, and `cache-dir` rows for diagnosing PATH and cache configuration; it now searches PATH using Windows path semantics and no longer flags a byte-identical copy of the binary as stale.

Diagnose PATH, binary identity, and cache location on a Windows machine after installing the preview zip.

```
$ ripwire . --doctor
```

Verify a fresh Windows installation is healthy — every row should show `ok='1'` before running ripwire in CI.

```
$ ripwire . --doctor
```

— Names all diagnostic fields and the exact behavior fixed. v0.6.5

03 **TypeScript path-alias imports resolved as real edges** IMPROVED

80

ripwire now resolves TypeScript alias imports (tsconfig path aliases) as real call-graph edges, so `ripwire . --impact src/utils/auth.ts` sees the full downstream blast radius for modules imported via an alias rather than missing them.

Trace the blast radius of a TypeScript module that is imported via a tsconfig path alias — now that alias imports are real edges, `--impact` sees the full downstream graph.

```
$ ripwire . --impact src/utils/auth.ts
```

— Names mechanism and gives exact --impact command demonstrating it. v0.6.5

04 **Cache warm-up diagnostics via RIPWIRE_CACHE_STATS** NEW

75

Setting `RIPWIRE_CACHE_STATS=1` prints a `cache-stats` line on stderr with `reparsed=`, `reused=`, `files=`, and `warm_growths=` counters, letting you confirm a run is fully cache-warm (`reparsed=0`) before relying on it in CI.

Verify cache is fully warm after a second run — every file should be served from cache with `reparsed=0`.

```
$ RIPWIRE_CACHE_STATS=1 ripwire . > /dev/null
```

Confirm every file is served from cache on a warm run — useful before committing to a CI caching strategy.

```
$ RIPWIRE_CACHE_STATS=1 ripwire . > /dev/null
```

— Names exact env var and counters, includes runnable command. product docs

05 **Rank functions by Halstead complexity** IMPROVED

60

`ripwire . --biggest-first` ranks functions largest-first by Halstead volume, useful for triaging the highest-complexity code or prioritising which changed functions to review — the score is not a readability grade.

Rank functions largest-first by Halstead volume to triage the highest-complexity code in a repo.

```
$ ripwire . --biggest-first
```

Rank functions largest-first by Halstead volume to prioritise which changed functions to review, without treating the score as a readability grade.

```
$ ripwire . --biggest-first
```

— Runnable command shown but no detail on how Halstead volume is computed. product docs

THINNER COVERAGE BELOW

06 **--for lifts verbatim-named symbols** IMPROVED 40

`--for` now lifts a symbol named verbatim in the task, improving relevance of call-graph results.

— One-line description, no example or before/after detail. v0.6.5

07 **Discloses unreadable configs** IMPROVED 20

ripwire now discloses configs that it cannot read rather than silently guessing their contents.

— Thin one-line description with no named surface or example. v0.6.5

v0.6.5 NOTES ↗ LATEST STABLE

ripwire v0.6.5 resolves TypeScript alias imports as real edges and adds `declined_calls=` counting for ambiguous method bindings.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.6.5 https://github.com/redhat-et/ripwire.git
# already have the repo? check out this version:
$ git checkout v0.6.5
```

└──▷ TRY IT

Check which calls ripwire declined to bind — useful when auditing a Python or TS codebase where dict/map method calls are widespread and you want to know what coverage is missing.

```
$ ripwire . --for="process_payment" 2>&1 | grep declined_calls=
```

Trace the blast radius of a TypeScript module that is imported via a tsconfig path alias — now that alias imports are real edges, `--impact` sees the full downstream graph.

```
$ ripwire . --impact src/utils/auth.ts
```

Adds `declined_calls=` output field in Python, JS/TS and Ruby when a method call on a dict or map cannot be bound to a specific in-repo method because the calling file does not name the class (or one in its inheritance cone).

`--for` now lifts a symbol named verbatim in the task, improving relevance of call-graph results.

`--doctor` on Windows now searches PATH using Windows path semantics and no longer flags a byte-identical copy as stale.

Discloses configs that ripwire cannot read rather than silently guessing their contents.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

GPT Researcher

Feedback ⓘ v3.7.0 NOTES ↗ RANK

GPT Researcher runs autonomous research workflows over web and local sources and produces cited reports.

> GPT Researcher expanded its retriever ecosystem with six new search integrations and multi-retriever combination, while overhauling its context-filtering pipeline to use Jev-based relevance scoring with a BM25 fallback and dropping support for Python versions before 3.12.

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

01 Six new search retrievers and multi-source combination NEW

90

Adds Brave Search via `RETRIEVER=brave` and `BRAVE_API_KEY`; OpenAlex via `RETRIEVER=openalex` with optional `OPENALEX_EMAIL` and `OPENALEX_API_KEY` for better rate limits; Semantic Scholar via `RETRIEVER=semantic_scholar`; fastCRW via `RETRIEVER=crw`; MCP via `RETRIEVER=mcp`; and PubMed Central full-text retrieval via optional `NCBI_API_KEY` and `PUBMED_DB` (default `pmc`). Multiple retrievers can now be combined in a single run, e.g. `RETRIEVER=tavily,openalex,semantic_scholar`, for academic literature reviews.

Run an academic literature review by combining a broad web retriever with two scholarly sources in a single research session.

```
$ RETRIEVER=tavily,openalex,semantic_scholar python -m
gpt_researcher
```

Configure OpenAlex with an optional email and API key to avoid rate-limiting during large-scale scholarly research runs.

```ini
RETRIEVER=openalex
OPENALEX_EMAIL=you@example.com
OPENALEX_API_KEY=your_api_key_here
```

Enable PubMed Central full-text retrieval with an optional NCBI API key to increase rate limits for biomedical research.

```ini
NCBI_API_KEY=your_api_key_here
PUBMED_DB=pmc
```

— Names every retriever flag and env var with runnable examples `product docs`

02 **Jev-based context filtering with BM25 fallback** BREAKING

85

`TYPESAFE_API_KEY` enables Jev usefulness-based scoring for context filtering, replacing embedding similarity with relevance ranking; it falls back to BM25 keyword ranking when the key is unset or a Jev call fails, so an embeddings provider is no longer required. The default context filter is now BM25 instead of embeddings; set `CONTEXT_FILTER=embeddings` to restore the previous behaviour.

Enable Jev usefulness-based context filtering to improve research report quality without needing a local embeddings provider.

```
$ export TYPESAFE_API_KEY=your_typesafe_key
```

— Explains mechanism and before/after default change `v3.7.0`

03 **Configurable Serper retriever endpoint** NEW

60

`SERPER_API_BASE` environment variable makes the Serper retriever's base URL configurable, allowing use of self-hosted or alternative Serper endpoints.

Point the Serper retriever at a self-hosted or alternative search endpoint instead of the default.

```
$ export SERPER_API_BASE=https://your-serper-instance.example.com
```

— Names exact env var with runnable example, thin explanation `v3.7.0`

THINNER COVERAGE BELOW

04 **Third-party retriever plugins via entry points** NEW

50

Search providers can now ship as their own installable package registered through the `gpt_researcher.retrievers` entry point, enabling third-party retriever plugins.

— Names the entry point but no usage example `v3.7.0`

05 **Python 3.12 now required** BREAKING 35

GPT Researcher now requires Python 3.12 or later; earlier Python versions are no longer supported.

— Clear requirement but no migration guidance v3.7.0

06 **Ground-truth-free report quality evaluation** NEW 25

Adds a `quality_eval` module for evaluating research report quality without requiring ground-truth data.

— Bare module name, no detail on usage or metrics v3.7.0

└──▷ **BREAKING ON UPGRADE**

! Python 3.12 or later is now required; earlier Python versions are no longer supported.

! The default context filter is now BM25 keyword ranking instead of embeddings. Set `CONTEXT_FILTER=embeddings` to restore the previous behaviour.

v3.7.0 NOTES ↗ LATEST STABLE

GPT Researcher v3.7.0 adds Jev-based context filtering, keyword fallback, and third-party retriever plugins via entry points.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v3.7.0 https://github.com/assafelovic/gpt-researcher.git
# already have the repo? check out this version:
$ git checkout v3.7.0
```

└──▷ **TRY IT**

Enable Jev usefulness-based context filtering to improve research report quality without needing a local embeddings provider.

```
$ export TYPESAFE_API_KEY=your_typesafe_key
```

Point the Serper retriever at a self-hosted or alternative search endpoint instead of the default.

```
$ export SERPER_API_BASE=https://your-serper-instance.example.com
```

Adds `TYPESAFE_API_KEY` environment variable to enable Jev-based usefulness scoring for context filtering, replacing embedding similarity with relevance ranking;

falls back to BM25 keyword ranking when unset or when a Jev call fails, making an embeddings provider no longer required.

Adds `CONTEXT_FILTER=embeddings` environment variable to restore the previous embeddings-based context filter behaviour when the BM25 keyword default is not desired.

Adds `SERPER_API_BASE` environment variable to make the Serper retriever base URL configurable for self-hosted or alternative endpoints.

Supports shipping a search provider as its own installable package via the `gpt_researcher.retrievers` entry point, enabling third-party retriever plugins.

Adds `quality_eval` module for ground-truth-free report quality evaluation.

└─▷ **BREAKING ON UPGRADE**

! Python 3.12 or later is now required; earlier Python versions are no longer supported.

! The default context filter is now BM25 keyword ranking instead of embeddings. Set `CONTEXT_FILTER=embeddings` to restore the previous behaviour.

WAS THIS USEFUL?  

Feynman

Feedback ⓘ v0.5.9 NOTES ↗ RANK

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

> Feynman's latest releases pass Pi's session-management flags straight through, gate a new alphaXiv research-tool suite behind login with automatic fallback to Semantic Scholar and OpenAlex, and clean up a batch of internal config paths and legacy commands.

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

01 Pi session flags passthrough NEW 85

Feynman now passes Pi's session management flags through instead of rejecting them: `--continue` / `-c`, `--resume` / `-r`, `--session <path|id>`, `--fork <path|id>`, `--no-session`, and `--export <session.jsonl> [out.html]`.

Resume a previous research session by ID when starting a new Feynman run, now that Pi session flags pass through.

```
$ feynman --resume <session-id>
```

— All flags named plus a runnable resume example product docs

02 AlphaXiv research tools with auth gating NEW 85

Signing in with `feynman alpha login` (checked via `feynman alpha status`) registers the alpha agent tools `alpha_search`, `alpha_get_paper`, `alpha_ask_paper`, and `alpha_read_code` for the next session; unauthenticated sessions skip these and fall back automatically to Semantic Scholar, OpenAlex, and other literature databases.

Authenticate with alphaXiv on a headless server (SSH/Docker/WSL) to unlock the alpha agent tools in your next session.

```
$ feynman alpha login
```

After signing in, check alphaXiv auth status before starting a new session to confirm the agent tools will be available.

```
$ feynman alpha status
```

Authenticate with alphaXiv so that `alpha_search`, `alpha_get_paper`, `alpha_ask_paper`, and `alpha_read_code` are registered for your session — then start a new session to pick them up.

```
$ feynman alpha login
```

— Names tools, fallback providers and exact commands v0.5.9

---

03 **Usage telemetry with opt-out** NEW 80

Adds telemetry events `feynman_session_started`, `feynman_workflow_started`, `feynman_workflow_completed`, `feynman_tool_used`, and one PostHog `$ai_generation` event per model response; set `DO_NOT_TRACK=1` to opt out, and run `feynman status` to see whether telemetry is on.

— Named events, env var and status command given product docs

---

04 **ACP server mode for editor integration** NEW 70

Feynman can run as an ACP server so editors like Zed can use it as their AI backend, launched with `PI_ACP_PI_COMMAND=feynman feynman --mode rpc --no-themes`.

Run Feynman as an ACP server so Zed (or another ACP editor) can use it as its AI backend.

```
$ PI_ACP_PI_COMMAND=feynman feynman --mode rpc --no-themes
```

— Runnable command given but no further mechanism explained product docs

---

05 **Package installs routed through Pi** IMPROVED 70

`feynman packages install` now routes through Pi's own `pi update --extensions` / `pi install` commands, installing optional packages into `~/.feynman/agent/npm`.

— Commands and install path named, no example run product docs

---

06 **Foreground subagent research tool availability fix** IMPROVED 65

Points pi-subagents' `defaultSubagentOnlyExtensions` at `web_search`, `fetch_content`, and `feynman_science_database_search` so foreground `researcher` and `verifier` subagents have their research tools available.

— Config key named but no direct user-facing action v0.5.9

---

07 **Web-search config path moved** BREAKING 65

Web-search configuration now lives at `~/.feynman/agent/web-search.json`, where pi-web-access reads it; an existing `~/.feynman/web-search.json` is automatically migrated there on first launch.

— Exact paths given, migration behavior explained `product docs`

---

08 **Subagent directory exclusion default** IMPROVED 60

Sets `subagents.agentExcludeDirs` to `['~/.agents']` so agent files placed there no longer silently replace Feynman's bundled agents.

— Config key and default named, brief mechanism `product docs`

---

09 **FEYNMAN_HOME path expansion fix** BREAKING 60

`FEYNMAN_HOME=~` and `~/path` now expand to the real home folder, replacing the previous behavior of creating a literal `~` directory; existing setups that relied on the literal behavior may see paths change.

— Before/after behavior stated, migration note implicit `product docs`

└──▷ ALSO FROM THESE RELEASES

Set a larger context window and token limit for a local Ollama model so it can write files during research runs.

```json
{
  "contextWindow": 32768,
  "maxTokens": 8192
}
```

└──▷ BREAKING ON UPGRADE

! The `geminiBrowser`, `allowBrowserAuth`, `browserAuth`, and `chromeProfile` keys are no longer read; set `allowBrowserCookies` and `browserCookies.profile` in their place to retain Gemini Web cookie access.

! Web-search config is now read from `~/.feynman/agent/web-search.json`; `~/.feynman/web-search.json` is moved there on first launch.

! The `/feynman-model` command is removed; use Pi's `/model` for the main model and `subagents.agentOverrides.<name>.model` for per-subagent model selection.

! The `session-search` optional package preset is removed; search past sessions with `grep` or `find` over `~/.feynman/sessions`.

! The sealed runtime workspace (`.feynman/runtime-workspace.tgz`, its hashes, and its lock) is removed and no longer re-verified on launch.

! `FEYNMAN_HOME=~` previously created a literal `~` folder; it now expands to the real home directory, which may change paths on existing setups that relied on the literal behavior.

v0.5.9 NOTES ↗ LATEST STABLE

Feynman v0.5.9 fixes foreground subagent tool availability and gates alphaXiv tools behind authentication.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.5.9 https://github.com/Companion-Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.9
```

└──▷ **TRY IT**

Authenticate with alphaXiv so that `alpha_search`, `alpha_get_paper`, `alpha_ask_paper`, and `alpha_read_code` are registered for your session — then start a new session to pick them up.

```
$ feynman alpha login
```

Registers `alpha_search`, `alpha_get_paper`, `alpha_ask_paper`, and `alpha_read_code` tools only when signed in via `feynman alpha login`; unsigned sessions fall back to Semantic Scholar, OpenAlex, and other literature databases automatically.

Points pi-subagents' `defaultSubagentOnlyExtensions` at `web_search`, `fetch_content`, and `feynman_science_database_search` so foreground `researcher` and `verifier` subagents have their research tools available.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Google Gemini Enterprise Agent Platform

Feedback ⓘ changelog · Sep 27, 2026 CHANGELOG ↗ RANK

Gemini Enterprise Agent Platform provides Google's managed model training, inference, evaluation, and agent deployment services, formerly offered under Vertex AI.

> This window's changes center on CodeMender v0.10.0, which adds PR-focused diff scanning, SARIF export with CI gating, an adaptive deep-scan mode, an HTML report dashboard, and a configurable home directory, alongside a breaking SDK repackaging and two new models joining the platform.

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

01 **PR diff scanning in CodeMender** NEW 85

Adds `--diff` and `--staged` flags to `cm find` to audit only modified pull request hunks and 1-hop neighbor files, while suppressing pre-existing vulnerabilities outside the changed scope.

Scan only the changed hunks in a PR and export results as SARIF, failing CI if any HIGH or CRITICAL findings are present.

```
$ cm find --diff --staged && cm report --format sarif --output results.sarif --fail-on high
```

— Named flags, clear mechanism, runnable example given changelog-20260927-f87dee2c

02 **SARIF export and severity CI gating** NEW 85

Adds `--format sarif` (with `--output`) and `--fail-on` to `cm report` for SARIF v2.1.0 export and severity-based CI gating, e.g. failing a build on HIGH or CRITICAL findings.

Scan only the changed hunks in a PR and export results as SARIF, failing CI if any HIGH or CRITICAL findings are present.

```
$ cm find --diff --staged && cm report --format sarif --output results.sarif --fail-on high
```

— Exact flags, format version, and runnable CI command changelog-20260927-f87dee2c

03 **Adaptive deep scan mode** NEW 85

Adds `--deep` and `--deep-workers` flags to `cm find` for adaptive hybrid deep scanning combining deterministic file pruning, parallel package-batched analysis, and post-triage verification.

Run a deep parallel scan for higher vulnerability recall on a large repository, using 8 worker processes.

```
$ cm find --deep --deep-workers 8
```

— Mechanism spelled out with runnable worker-count example `changelog-20260927-f87dee2c`

---

04 **CM_HOME environment variable** NEW 70

Adds support for the `CM_HOME` environment variable to override the default `~/.codemender` directory for configuration, `state.db`, backups, and artifacts.

Redirect all CodeMender state, config, and artifacts to a shared team directory instead of the per-user default.

```
$ export CM_HOME=/opt/codemender/shared
cm find
```

— Named env var and paths, with usage example `changelog-20260927-f87dee2c`

---

05 **Interactive HTML report dashboard** NEW 65

Adds an interactive HTML dashboard to `cm report --format html` with severity metric cards, syntax-highlighted code snippets, and an inline patch diff viewer, plus `--open` (`-o`) to launch the report in the default browser.

— Flags and UI contents named, no example command `changelog-20260927-f87dee2c`

└──▷ **BREAKING ON UPGRADE**

! The `google-cloud-agentplatform` v2.0.1 SDK decouples the agent surface from `google-cloud-aiplatform` into a dedicated package with restructured namespaces — existing code importing from `google-cloud-aiplatform` agent modules requires migration per the Agent Platform SDK for Python version 2.0.1 migration guide.

`changelog · Sep 27, 2026` CHANGELOG ↗ LATEST STABLE

CodeMender v0.10.0 adds SARIF export, PR delta scanning, and adaptive deep scan; Gemini 3.8 Flash Cyber and Muse Spark 1.3 join the platform.

└──▷ **TRY IT**

Scan only the changed hunks in a PR and export results as SARIF, failing CI if any HIGH or CRITICAL findings are present.

```
$ cm find --diff --staged && cm report --format sarif --output
results.sarif --fail-on high
```

Run a deep parallel scan for higher vulnerability recall on a large repository, using 8 worker processes.

```
$ cm find --deep --deep-workers 8
```

Redirect all CodeMender state, config, and artifacts to a shared team directory instead of the per-user default.

```
$ export CM_HOME=/opt/codemender/shared
cm find
```

Adds `--diff` and `--staged` flags to `cm find` to audit only modified pull request hunks and 1-hop neighbor files while suppressing pre-existing vulnerabilities.

Adds `--format sarif` (`--output`) and `--fail-on` to `cm report` for SARIF v2.1.0 export and severity-based CI gating.

Adds `--deep` and `--deep-workers` flags to `cm find` for adaptive hybrid deep scanning combining deterministic file pruning, parallel package-batched analysis, and post-triage verification.

Adds support for the `CM_HOME` environment variable to override the default `~/.codemender` directory for configuration, `state.db`, backups, and artifacts.

Adds interactive HTML dashboard to `cm report --format html` with severity metric cards, syntax-highlighted code snippets, and an inline patch diff viewer, plus `--open` (`-o`) to launch the report in the default browser.

└──▷ **BREAKING ON UPGRADE**

! The `google-cloud-agentplatform` v2.0.1 SDK decouples the agent surface from `google-cloud-aiplatform` into a dedicated package with restructured namespaces — existing code importing from `google-cloud-aiplatform` agent modules requires migration per the Agent Platform SDK for Python version 2.0.1 migration guide.

WAS THIS USEFUL? 👍 👎

Hugging Face

Feedback ⓘ changelog · Sep 27, 2026 CHANGELOG ↗

RANK

Hugging Face is an open platform for building, sharing, and deploying machine-learning models, datasets, and applications.

> Hugging Face added episode preview playback for LeRobot datasets and enabled organizations to subscribe to Hugging Face via the Google Cloud Marketplace for consolidated billing.

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

01 Google Cloud Marketplace billing subscription NEW

70

Organizations can subscribe to Hugging Face (Inference Providers, Spaces, storage, and Team or Enterprise seats) via the Google Cloud Marketplace, consolidating usage onto an existing Google Cloud invoice. Setup involves subscribing from the Marketplace listing, authenticating with a Google Cloud account, and linking the Hugging Face organization so usage appears automatically on the GCP invoice.

Subscribe an organization to Hugging Face through Google Cloud Marketplace to route all Inference Provider, Spaces, and seat costs to an existing GCP invoice.

📍 1. Go to the Google Cloud Marketplace listing for Hugging Face. 2. Click Subscribe and authenticate with your Google Cloud account. 3. Select the Hugging Face organization to link. 4. Wait for subscription validation — usage then appears on your Google Cloud invoice automatically.

— Includes concrete step-by-step setup instructions changelog-20260927-06c3565d

changelog · Sep 27, 2026 CHANGELOG ↗ LATEST STABLE

Hugging Face: Adds episode previews for LeRobot datasets on the dataset page, with synchronized multi-camera playback on a shared transport and hover previews in the episode list.

└──▷ **HOW TO FIND IT**

Subscribe an organization to Hugging Face through Google Cloud Marketplace to route all Inference Provider, Spaces, and seat costs to an existing GCP invoice.

📍 1. Go to the Google Cloud Marketplace listing for Hugging Face. 2. Click Subscribe and authenticate with your Google Cloud account. 3. Select the Hugging Face organization to link. 4. Wait for subscription validation — usage then appears on your Google Cloud invoice automatically.

> Dataset listings for LeRobot datasets now surface robot type, episode count, and a first-episode preview before opening the dataset.
>
> Enables organizations to subscribe to Hugging Face (Inference Providers, Spaces, storage, and Team or Enterprise seats) via the Google Cloud Marketplace, consolidating usage onto an existing Google Cloud invoice.

WAS THIS USEFUL? 👍 👎

Perplexity API

Feedback ⓘ snapshot-20260927 NOTES ↗ RANK

Perplexity API Platform - build with the Router, Agent, Search, and Embeddings APIs. Real-time, web-wide research and Q&A capabilities for your products.

> Perplexity API added a faster, cheaper Search API mode and rolled out GPT-6 model support across the Agent and Router APIs, while setting a hard deadline to retire several legacy OpenAI model IDs.

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

01 Search API search_type parameter for fast vs. web search NEW

80

Adds `search_type: "web"` as the explicit value to request standard web search on the Search API and Agent API, alongside a new `search_type: "fast"` option that cuts latency by roughly 800 ms and reduces cost to $1.00 per 1,000 requests compared to standard web search.

Run a Search API query with Fast Search to cut latency by ~800 ms and halve search costs compared to standard web search.

```
$ curl https://api.perplexity.ai/search \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"query": "latest CVE disclosures this week",
"search_type": "fast"}'
```

Use Fast Search in a Search API request to cut latency by ~800 ms and reduce cost to $1.00/1k requests — useful for high-volume, time-sensitive threat-intel lookups.

```
$ curl -X POST https://api.perplexity.ai/search \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"query": "latest CVEs in OpenSSL", "search_type": "fast"}'
```

— Names exact parameter values, latency and price, with runnable curl examples

snapshot-20260927

└──▷ **BREAKING ON UPGRADE**

! On 2026-10-24 at 00:00 UTC, the Agent API and Router API will stop accepting model IDs `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`,

`openai/gpt-5.1`, `openai/gpt-5`, and `openai/gpt-5-mini`; retired IDs will no longer be accepted or returned by model-list endpoints.

! On October 24, 2026 at 00:00 UTC, the Agent API and Router API will stop accepting the following model IDs: `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-5`, and `openai/gpt-5-mini` — direct model selections and fallback chains referencing these IDs must be updated before the cutoff.

`snapshot-20260927` NOTES ↗ **LATEST** STABLE

Perplexity Agent API gains GPT-6 Sol/Luna, Claude Opus 5.5, Grok 4.7, and a new Fast Search path at $1.00/1k requests.

└──▷ **TRY IT**

Use Fast Search in a Search API request to cut latency by ~800 ms and reduce cost to $1.00/1k requests — useful for high-volume, time-sensitive threat-intel lookups.

```
$ curl -X POST https://api.perplexity.ai/search \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"query": "latest CVEs in OpenSSL", "search_type": "fast"}'
```

Switch an Agent API workflow to `openai/gpt-6-sol` or `anthropic/claude-opus-5-5` for more capable reasoning on complex investigation tasks.

```
$ curl -X POST https://api.perplexity.ai/agent \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "anthropic/claude-opus-5-5", "messages": [{"role":
"user", "content": "Summarize recent threat actor TTPs targeting
financial sector APIs."}]}'
```

Updates the Agent API `low` and `medium` presets from `openai/gpt-5.6-luna` to `openai/gpt-6-luna`, and the `high` preset from `openai/gpt-5.6-sol` to `openai/gpt-6-sol`.

└──▷ **BREAKING ON UPGRADE**

! On October 24, 2026 at 00:00 UTC, the Agent API and Router API will stop accepting the following model IDs: `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-5`, and `openai/gpt-5-mini` — direct model selections and fallback chains referencing these IDs must be updated before the cutoff.

WAS THIS USEFUL?  

Eigen Labs Darkbloom

Feedback ⓘ v0.9.10 NOTES ↗

CODE ↗ RANK

Private Inference Network on Idle Macs

> Darkbloom's v0.9.10 release lets operators hot-swap models on running provider nodes without downtime, adds explicit model-cache location management, and overhauls provider reputation reporting into raw job counts.

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

01 **Explicit model-cache location management** NEW 90

`darkbloom models location` subcommand manages where model weights are stored, with `--from-env` to import the currently-exported Hugging Face location, `--check [path]` to verify a path, and `--reset` to clear the setting; existing caches are not automatically migrated when the location changes.

Pin your model-weight cache to a specific directory so ambient Hugging Face/XDG variables never redirect downloads or discovery.

```
$ darkbloom models location /Volumes/FastSSD/model-cache
```

Import the currently-exported Hugging Face location once and lock it so later environment changes cannot redirect it.

```
$ darkbloom models location --from-env
```

— Names subcommand and all flags with runnable examples v0.9.10

02 **Live model switching via `darkbloom switch`** NEW 85

The new `darkbloom switch` subcommand replaces a running provider's hosted model selection after a graceful drain, without requiring a process restart, coordinator reconnect, or re-attestation.

Hot-swap the model your provider node is serving mid-session — no restart or re-attestation needed.

```
$ darkbloom switch
```

— Names exact subcommand and mechanism with runnable example v0.9.10

03 **Provider reputation replaced with job counts** BREAKING

60

Replaces provider reputation scores with total, successful, and failed job counts; removes the composite score calculation and owner API field, so historical job failures no longer imply reduced routing priority in the dashboard.

— Clear before/after but no field names or endpoint given v0.9.10

THINNER COVERAGE BELOW

04 **App Attest failure diagnostics** IMPROVED 50

Adds App Attest diagnostics for unexplained App Attest failures, including detection and rotation of dead Secure Enclave keys that previously locked providers out until manual keychain intervention.

— Explains mechanism but no command to invoke it v0.9.10

05 **Model preloading on provider start** IMPROVED 45

Darkbloom now preloads selected models on every provider start, including standalone `--local` mode, regardless of idle-memory policy.

— Describes behavior change but no configuration surface to act on v0.9.10

06 **Public model demand history with CSV export** NEW 25

Darkbloom adds public model demand history with CSV export.

— Only mentioned in summary, no further detail available v0.9.10

v0.9.10 NOTES ↗ CODE ↗ LATEST STABLE

Darkbloom v0.9.10 adds live model switching, a new model-cache location command, and public model demand history with CSV export.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.9.10 https://github.com/Layr-Labs/d-inference.git
# already have the repo? check out this version:
$ git checkout v0.9.10
```

└──▷ TRY IT

Hot-swap the model your provider node is serving mid-session — no restart or re-attestation needed.

```
$ darkbloom switch
```

Pin your model-weight cache to a specific directory so ambient Hugging Face/XDG variables never redirect downloads or discovery.

```
$ darkbloom models location /Volumes/FastSSD/model-cache
```

Import the currently-exported Hugging Face location once and lock it so later environment changes cannot redirect it.

```
$ darkbloom models location --from-env
```

Adds `darkbloom switch` subcommand to replace a running provider's hosted model selection after a graceful drain, without process restart, coordinator reconnect, or re-attestation.

Adds `darkbloom models location` subcommand with `--from-env`, `--check [path]`, and `--reset` flags to explicitly manage where model weights are stored, without automatic migration of existing caches.

Preloads selected models on every provider start, including standalone `--local` mode, regardless of idle-memory policy.

Adds App Attest diagnostics for unexplained App Attest failures, including detection and rotation of dead Secure Enclave keys that previously locked providers out until manual keychain intervention.

Replaces provider reputation scores with total, successful, and failed job counts; removes the composite score calculation and owner API field so historical job failures no longer imply reduced routing priority in the dashboard.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.79 NOTES ↗ RANK

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

> HyperFrames v0.8.79 adds animation-end reporting for sequencing and pre/post-change file snapshots in Studio Server History.

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

01 **Pre/post-change file snapshots in Studio Server History** NEW 44

Studio Server History can copy files before or after any change into a folder, enabling snapshot-based version checkpoints.

– Names feature area but no exact config or flag v0.8.79

02 **Animation-end reporting for compositions** NEW 35

The runtime now reports where a composition's animations end, one cycle each, giving agents and tooling a reliable signal for sequencing.

– Describes behaviour but no API name given v0.8.79

v0.8.79 NOTES ↗ LATEST STABLE

HyperFrames v0.8.79 adds animation-end reporting and pre/post-change file snapshots via History.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.8.79 https://github.com/heygen-
com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.79
```

The runtime now reports where a composition's animations end, one cycle each, giving agents and tooling a reliable signal for sequencing.

Studio Server History can copy files before or after any change into a folder, enabling snapshot-based version checkpoints.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

KoboldCpp

Feedback ⓘ v1.122.1 NOTES ↗ CODE ↗ RANK

KoboldCpp runs GGUF language models locally as a single-file application with a built-in web UI and API.

> KoboldCpp's single release in this window adds AGENTS.md and context compaction support to its Agent, custom MCP tool integration, a new autoswap threshold flag, and a breaking change to how koboldcpp.sh builds are optimized by default.

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

01 `--autoswapthreshold` flag for VRAM residency NEW

80

A new `--autoswapthreshold` flag prevents autoswap from triggering when small models are loaded, keeping them fully in VRAM rather than spilling to slower memory, e.g. `koboldcpp-linux-x64 --model my-model.gguf --autoswapthreshold 4096`.

Prevent autoswap from triggering when small models are loaded, keeping them fully in VRAM rather than spilling to slower memory

```
$ koboldcpp-linux-x64 --model my-model.gguf --autoswapthreshold
4096
```

— Runnable command with named flag and clear purpose v1.122.1

02 Custom MCP tools via `mcp.json` NEW

70

The KoboldCpp Agent can be extended with custom MCP tools by pointing it at an `mcp.json` server definition, for example configuring `mcpServers` entries with a `url` field.

Extend the KoboldCpp Agent with custom MCP tools by pointing it at an mcp.json server definition

```json
{
  "mcpServers": {
    "demo": {
      "url": "http://localhost:8000/mcp"
    }
  }
}
```

— Config file and schema shown, but no further mechanism detail v1.122.1

THINNER COVERAGE BELOW

03 AGENTS.md support and context compaction in Agent NEW

30

KoboldCpp now supports `AGENTS.md` files and context compaction within the KoboldCpp Agent.

— Named but not explained beyond the headline v1.122.1

04 Keepalive padding for tool and text streaming IMPROVED

30

Adds keepalive whitespace padding for tool streaming and text sync requests, consistent with existing image-gen behavior.

— States what changed but no usage detail or command v1.122.1

└──▷ BREAKING ON UPGRADE

! `koboldcpp.sh` now optimizes builds for the local platform by default; package maintainers who distribute builds must now set the environment variable `KCPP_PORTABLE=1` to produce portable binaries that run on systems other than the build host.

v1.122.1 NOTES ↗ CODE ↗ LATEST STABLE

KoboldCpp: Supports `AGENTS.md` and context compaction within the KoboldCpp Agent

└──▷ GET THIS VERSION

```
$ git clone --branch v1.122.1
https://github.com/LostRuins/koboldcpp.git
# already have the repo? check out this version:
$ git checkout v1.122.1
```

└──▷ TRY IT

Prevent autoswap from triggering when small models are loaded, keeping them fully in VRAM rather than spilling to slower memory

```
$ koboldcpp-linux-x64 --model my-model.gguf --autoswapthreshold 4096
```

Extend the KoboldCpp Agent with custom MCP tools by pointing it at an mcp.json server definition

mcp.json

```
{
  "mcpServers": {
    "demo": {
      "url": "http://localhost:8000/mcp"
    }
  }
}
```

Adds keepalive whitespace padding for tool streaming and text sync requests, consistent with existing image-gen behavior

└─▷ **BREAKING ON UPGRADE**

! `koboldcpp.sh` now optimizes builds for the local platform by default; package maintainers who distribute builds must now set the environment variable `KCPP_PORTABLE=1` to produce portable binaries that run on systems other than the build host.

WAS THIS USEFUL? 👍 👎

vMLX

Feedback ⓘ v1.6.68 NOTES ↗ RANK

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

> vMLX 1.6.68 adds native support for the JANGH model format with legacy JANGTQ v2 compatibility and routed-expert loading, alongside a set of opt-in kernel-level optimizations spanning JANGH, GLM, and MTP model paths.

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

01 **Native JANGH/JANGTQ2 model format support** NEW 68

Introduces the `JANGH` checkpoint format with legacy `JANGTQ v2` compatibility, enabling native kernel loading at startup. Adds `JANGTQ2` routed-expert loading with validated native kernels via the feat(runtime) path (`bc7382aa`), and reuses exact codebook fits at JANGH kernel startup to reduce initialization overhead.

— Names format, compatibility target, and commit path but no CLI usage. v1.6.68

THINNER COVERAGE BELOW

02 **Opt-in kernel optimizations for JANGH, GLM, and MTP** NEW 50

Adds opt-in expert-aligned Hadamard prefill tiles for JANGH models, opt-in fused H32 rotation into expert prefill for JANGH models, an opt-in bounded exact KDA substitution kernel for GLM models, and opt-in bounded forward completion diagnostics for MTP (multi-token prediction).

— Names each kernel and target model but no flags or benchmarks given. v1.6.68

v1.6.68 NOTES ↗ LATEST STABLE

vMLX 1.6.68 introduces native JANGH model support with legacy JANGTQ v2 compatibility and routed-expert loading via validated native kernels.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.6.68 https://github.com/jjang-ai/vmlx.git
# already have the repo? check out this version:
$ git checkout v1.6.68
```

Introduces `JANGH` checkpoint format with legacy JANGTQ v2 compatibility, enabling native kernel loading at startup.

> Adds `JANGTQ2` routed-expert loading with validated native kernels via the feat(runtime) path (`bc7382aa`).
>
> Adds opt-in expert-aligned Hadamard prefill tiles for JANGH models.
>
> Adds opt-in fused H32 rotation into expert prefill for JANGH models.
>
> Adds opt-in bounded exact KDA substitution kernel for GLM models.
>
> +2 more

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

UK AI Security Institute Inspect AI

Feedback ⓘ

0.3.271 CODE ↗ RANK

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

> Inspect AI 0.3.270-0.3.271 added CLI flags for custom API gateway headers/body, an in-place top-up mode for growing eval_set datasets, and LiteLLM Proxy defaults enabling the OpenAI Responses API and richer Claude reasoning logs.

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

01 Responses API default for LiteLLM Proxy models IMPROVED

80

GPT-5, o-series, and Codex models served through the LiteLLM Proxy now default to the OpenAI Responses API with streaming and reasoning carried between turns; the new `-M responses_api=false` flag opts back out to the Chat Completions path.

Opt out of the new Responses API default when your LiteLLM Proxy setup requires the Chat Completions path for GPT-5 or o-series models.

```
$ inspect eval mytask.py --model litellm_proxy/gpt-5 -M
responses_api=false
```

— Named flag and models with runnable example 0.3.270

02 Custom headers and body via CLI flags NEW

75

`inspect eval` and `inspect eval-set` now accept `--extra-headers` and `--extra-body` options, letting users inject custom API gateway headers and extra body fields into every model request without modifying eval code.

Inject a custom API gateway header and extra body field into every model request from the CLI, without modifying eval code.

```
$ inspect eval mytask.py --extra-headers '{"X-My-Gateway-Token":
"abc123"}' --extra-body '{"custom_param": true}'
```

— Named CLI flags with a runnable example 0.3.271

03 **Claude reasoning logs via LiteLLM Proxy** IMPROVED 75

Claude 4.7+ models accessed via the LiteLLM Proxy now record summarized thinking in the log even through older LiteLLM versions that previously returned none; the new `-M thinking_display=omitted` flag suppresses this to keep logs compact.

Suppress summarized thinking output in logs when evaluating Claude 4.7+ through the LiteLLM Proxy to keep logs compact.

```
$ inspect eval mytask.py --model litellm_proxy/claude-4-7 -M
thinking_display=omitted
```

— Named flag with before/after behaviour and example 0.3.270

04 **In-place top-ups for eval_set** IMPROVED 65

`eval_set` can now top up a grown dataset in place — reusing prior samples and running only newly added samples — when the dataset is a strict superset with stable sample ids; a shrunk dataset or unstable ids still triggers a full re-run.

— Clear mechanism and conditions but no direct example 0.3.270

THINNER COVERAGE BELOW

05 **Google token counting fallback estimate** IMPROVED 55

Google provider token counting now falls back to a local tiktoken-based estimate (with a one-time warning) when the `countTokens` endpoint is unavailable or returns a non-retryable error, instead of failing the sample.

— Explains mechanism but no user-facing action 0.3.271

0.3.271 CODE ↗ LATEST STABLE

Inspect AI 0.3.271 adds `--extra-headers` and `--extra-body` CLI options to `inspect eval` and `inspect eval-set`.

└──▷ **GET THIS VERSION**

```
$ git clone --branch 0.3.271
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.271
```

└──▷ **TRY IT**

Inject a custom API gateway header and extra body field into every model request from the CLI, without modifying eval code.

```
$ inspect eval mytask.py --extra-headers '{"X-My-Gateway-Token":
"abc123"}' --extra-body '{"custom_param": true}'
```

Google provider token counting falls back to a local tiktoken-based estimate (with a one-time warning) when the `countTokens` endpoint is unavailable or returns a non-retryable error, instead of failing the sample.

0.3.270 CODE ↗ STABLE

Inspect AI 0.3.270 adds in-place eval_set top-ups, OpenAI Responses API default via LiteLLM Proxy, and richer Claude reasoning logs.

└──▷ **GET THIS VERSION**

```
$ git clone --branch 0.3.270
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.270
```

└──▷ **TRY IT**

Opt out of the new Responses API default when your LiteLLM Proxy setup requires the Chat Completions path for GPT-5 or o-series models.

```
$ inspect eval mytask.py --model litellm_proxy/gpt-5 -M
responses_api=false
```

Suppress summarized thinking output in logs when evaluating Claude 4.7+ through the LiteLLM Proxy to keep logs compact.

```
$ inspect eval mytask.py --model litellm_proxy/claude-4-7 -M
thinking_display=omitted
```

Adds `-M responses_api=false` opt-out flag for GPT-5, o-series, and Codex models served through the LiteLLM Proxy, which now default to the Responses API with streaming and reasoning carried between turns.

Adds `-M thinking_display=omitted` opt-out flag for Claude 4.7+ models via the LiteLLM Proxy, which now record summarized thinking in the log even through older LiteLLM versions that previously returned none.

> Enables `eval_set` to top up a grown dataset in place — reusing prior samples and running only newly added samples — when the dataset is a strict superset with stable sample ids; a shrunk dataset or unstable ids still triggers a full re-run.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ changelog · Sep 27, 2026 CHANGELOG ↗

RANK

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

> Braintrust's biggest addition this window is a new Debugger that diagnoses likely failure modes in agent traces, alongside a Pattern frequency view for recurring findings, Loop thread sharing, tracing for Grok and Google Antigravity coding sessions, SELECT DISTINCT query support, and tunable S3 attachment-upload settings — plus two breaking changes to provider model scoping and CLI topic-config flags.

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

01 Tracing for Grok and Google Antigravity coding sessions NEW

85

Adds tracing for Grok and Google Antigravity coding sessions via `bt` CLI (v0.19.3+), capturing each session as a Braintrust trace without interrupting the coding agent if tracing fails. A redaction plugin can strip API keys and PII from coding-agent spans before they are sent to Braintrust, e.g. `bt trace enable my-redact-plugin.js && bt trace run -- python my_agent.py`.

Strip API keys and PII from coding-agent spans before they are sent to Braintrust, ensuring sensitive data never leaves the machine.

```
$ bt trace enable my-redact-plugin.js && bt trace run -- python my_agent.py
```

— Includes a runnable CLI command sequence snapshot-20260927

02 Debugger for agent trace failure analysis NEW

75

Adds the Debugger, which reports likely failure modes for a single agent trace citing the specific spans, tool calls, and model outputs behind each, alongside an Analyze trace view that groups a run into Work sections explaining what it did; requires data plane v2.12.0+.

— Mechanism detailed but no exact UI navigation given snapshot-20260927

03 SELECT DISTINCT query support NEW

65

Braintrust now supports `SELECT DISTINCT` queries, letting users query a dataset for unique combinations of fields, such as model and score, without writing dedup logic in application code.

Query a dataset for unique combinations of model and score without writing dedup logic in application code.

```
$ SELECT DISTINCT model, score FROM logs WHERE project = 'my-project'
```

— Includes a runnable example query changelog-20260927-867ae438

THINNER COVERAGE BELOW

04 **Pattern frequency view for findings** NEW 55

Adds a Pattern frequency view showing how often a finding occurs among comparable traces or across a project, with a chart for the selected time range and scheduled refresh for existing active patterns; requires data plane v2.15.0+.

— UI capability described but no navigation path changelog-20260927-867ae438 snapshot-20260927

05 **Loop thread sharing** NEW 55

Adds Loop thread sharing so project members can view or chat in a shared Loop thread, with Loop responding using each sender's own permissions; requires data plane v2.15.0+.

— Permission mechanism explained, no navigation path snapshot-20260927

06 **RubyLLM 2.0 support in Ruby SDK** NEW 35

Adds support for `RubyLLM` 2.0 in the Ruby SDK.

— Names the library version with no further detail changelog-20260927-867ae438

07 **TypeSafe Jev scoring** NEW 15

Braintrust adds TypeSafe Jev scoring.

— Named only in summary, no further elaboration changelog-20260927-867ae438

08 **Span plugins** NEW 10

Braintrust adds span plugins.

— Named only, no mechanism or usage detail changelog-20260927-867ae438

09 **Project groups** NEW 10

Braintrust adds project groups.

— Named only, no further detail provided changelog-20260927-867ae438

└──▷ BREAKING ON UPGRADE

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0 — see the v0.20.0 migration instructions.

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

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

changelog · Sep 27, 2026 CHANGELOG ↗ LATEST STABLE

Braintrust adds OpenInference tracing, span plugins, TypeSafe Jev scoring, Loop thread sharing, project groups, and SELECT DISTINCT support.

└──▷ **TRY IT**

Query a dataset for unique combinations of model and score without writing dedup logic in application code.

```
$ SELECT DISTINCT model, score FROM logs WHERE project = 'my-project'
```

Adds `BRAINTRUST_ATTACHMENT_UPLOADER_QUEUE_SIZE`, `BRAINTRUST_ATTACHMENT_UPLOADER_REQUEST_TIMEOUT_MILLIS`, `BRAINTRUST_ATTACHMENT_UPLOADER_MAX_RETRIES`, and `BRAINTRUST_ATTACHMENT_UPLOADER_INITIAL_RETRY_DELAY_MILLIS` environment variables (and matching `BraintrustConfig` options) to tune S3 attachment uploader behavior, including automatic recovery after transient errors.

Adds support for `RubyLLM` 2.0 in the Ruby SDK.

Adds a Pattern frequency view showing how often a finding occurs among comparable traces or across a project, with a chart for the selected time range and scheduled refresh for existing active patterns.

└──▷ **BREAKING ON UPGRADE**

! The `--relabel-overlap` flag is removed from `bt topics config enable` and `bt topics config set`; it was deprecated and ignored since data plane v2.15.0 — see the v0.20.0 migration instructions.

! On Braintrust-hosted deployments, project providers now override organization providers only for models they explicitly serve; organization providers may serve

other models using their own credentials. If you relied on project providers to restrict model access or spending across all models, review your organization provider model selections and permissions.

snapshot-20260927 NOTES ↗ STABLE

Braintrust: Adds Pattern frequency view showing how often a finding occurs among comparable traces or across a project, with a chart for the selected time range; requires data plane v2.15.0+.

└──▷ TRY IT

Strip API keys and PII from coding-agent spans before they are sent to Braintrust, ensuring sensitive data never leaves the machine.

```
$ bt trace enable my-redact-plugin.js && bt trace run -- python my_agent.py
```

Adds the Debugger, which reports likely failure modes for a single agent trace citing the specific spans, tool calls, and model outputs behind each, alongside an `Analyze trace` view that groups a run into Work sections explaining what it did; requires data plane v2.12.0+.

Adds Loop thread sharing so project members can view or chat in a shared Loop thread, with Loop responding using each sender's own permissions; requires data plane v2.15.0+.

Adds tracing for Grok and Google Antigravity coding sessions via bt CLI (v0.19.3+), capturing each session as a Braintrust trace without interrupting the coding agent if tracing fails.

└──▷ BREAKING ON UPGRADE

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

WAS THIS USEFUL? 👍 👎

LangChain LangSmith

Feedback ⓘ changelog · Sep 27, 2026

CHANGELOG ↗ RANK

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

> LangSmith's window centers on new observability and access-control surfaces: a `/v2/threads/stats` endpoint, agent-scoped evaluator filtering, a prebuilt code-evaluator sandbox snapshot, and tighter permission requirements on dataset exports and deployment rollbacks.

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

01 **Evaluators API filtered by agent** NEW 75

A new query parameter lets you retrieve only the evaluators attached to a specific agent's environments or tagged datasets: `GET https://api.smith.langchain.com/api/v1/evaluators?agent_id=<agent_id>` with an `X-API-Key` header.

Retrieve only the evaluators attached to a specific agent's environments or tagged datasets, scoped to that agent.

```
$ curl -X GET 'https://api.smith.langchain.com/api/v1/evaluators?agent_id=<agent_id>' \
  -H 'X-API-Key: <your_api_key>'
```

— Runnable curl command with named endpoint and parameter changelog-20260927-d394ac7c

02 **Prebuilt code-evaluator sandbox snapshot** NEW 70

A new system sandbox snapshot, `system/code-evaluators:latest`, comes with NumPy, pandas, SciPy, and scikit-learn preinstalled for running code evaluations; create it from Sandboxes > New Sandbox in the LangSmith console.

Create a sandbox from the prebuilt evaluator snapshot to run code evaluations with NumPy, pandas, SciPy, and scikit-learn already installed.

> 📍 In the LangSmith console, go to Sandboxes › New Sandbox, select `system/code-evaluators:latest` from the system snapshot catalog, then click Create.

— Named snapshot and library list, exact UI navigation given changelog-20260927-d394ac7c

03 **Provider metadata for custom Gateway endpoints** NEW 70

Setting `ls_provider` (e.g. `"ls_provider": "openai"`) alongside `model` and `base_url` on a custom OpenAI-compatible endpoint attaches provider metadata in the LLM Gateway and enables correct pricing.

Attach provider metadata and enable correct pricing in the LLM Gateway for a custom OpenAI-compatible endpoint by setting `ls_provider`.

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

— Named field with concrete JSON config example `snapshot-20260927`

---

04 **Dataset export requires download permission** BREAKING

60

Dataset CSV and JSONL exports now require both dataset read and download permissions; API integrations that previously succeeded without the download permission will now receive a forbidden response.

— Names formats and permission change with before/after behavior `snapshot-20260927`

THINNER COVERAGE BELOW

05 **IAM role auth for ECR image references** IMPROVED 50

Pasting an ECR repository or image URL keeps AWS IAM role-based authentication available, so ECR-hosted images can be referenced without switching to static credentials.

— Describes the behavior but no config example shown `snapshot-20260927` `changelog-20260927-d394ac7c`

---

06 **Sandbox delegation grant restriction** BREAKING 40

A sandbox operating under an access-delegation grant can no longer grant a delegation of its own to sandboxes it creates.

— States the rule but no mechanism or workaround given `changelog-20260927-d394ac7c`

└──▷ BREAKING ON UPGRADE

! Scale-to-zero deployments reject rollbacks to revisions below LangGraph API 0.13.0 or without a known compatible version; the rollback dialog directs you to rebuild the target commit with a supported version.

! A sandbox operating under an access-delegation grant can no longer grant a delegation of its own to sandboxes it creates.

! Dataset CSV and JSONL exports now require both dataset read and download permissions; any existing API integration that lacked the download permission and previously succeeded will now receive a forbidden response.

changelog · Sep 27, 2026 CHANGELOG ↗ LATEST STABLE

LangSmith: Adds `Cache Hit %` to the project stats sidebar, showing the share of input tokens served from the prompt cache.

└──▷ **TRY IT**

Retrieve only the evaluators attached to a specific agent's environments or tagged datasets, scoped to that agent.

```
$ curl -X GET 'https://api.smith.langchain.com/api/v1/evaluators?
agent_id=<agent_id>' \
  -H 'X-API-Key: <your_api_key>'
```

Create a sandbox from the prebuilt evaluator snapshot to run code evaluations with NumPy, pandas, SciPy, and scikit-learn already installed.

📍 In the LangSmith console, go to Sandboxes › New Sandbox, select `system/code-evaluators:latest` from the system snapshot catalog, then click Create.

Pasting an ECR repository or image URL keeps AWS IAM role authentication available.

└──▷ **BREAKING ON UPGRADE**

! Scale-to-zero deployments reject rollbacks to revisions below LangGraph API 0.13.0 or without a known compatible version; the rollback dialog directs you to rebuild the target commit with a supported version.

! A sandbox operating under an access-delegation grant can no longer grant a delegation of its own to sandboxes it creates.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

FailproofAI

Feedback ⓘ v1.0.8 NOTES ↗ RANK

Observability and enforcement for AI agent harnesses. Capture every run and runtime reliability with policy enforcement.

> FailproofAI shipped Jev, a semantic classifier layer with its own CLI, policy-authority model, pack API, and Cloud/dashboard integration for safely clearing reviewable policy blocks, alongside a new TypeScript telemetry SDK, sandboxed server-side evaluations, human-input event tracking, and expanded `fp issues` lifecycle commands.

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

01 **Jev semantic classifier CLI (setup, status, test, models)**

NEW 95

Adds `failproofai jev setup --provider <typesafe|openrouter|vercel|cloudflare|custom>` (or `failproofai jev --url <url> --token <token>`) to configure Jev with a bring-your-own key, writing config to `~/.failproofai/jev.json` at mode 0600. Adds `failproofai jev status` to show provider, mode, how many policies Jev may clear, and the last 24 hours of evaluations, fallback rate, latency, and cleared calls; `failproofai jev test` fires one real evaluation and exits non-zero if the answer is too slow or wrong; `failproofai jev models` lists the models an endpoint serves and marks the one in use; and a `mode: off` option keeps config in place while stopping evaluations.

Check Jev health and last-24-hour metrics — fallback rate, latency, and which policies it cleared — after enabling enforce mode.

```
$ failproofai jev status
```

— names every subcommand, flag and config path v1.0.8

02 **Issue lifecycle subcommands in `fp issues`** NEW 95

Adds `close`, `archive`, `unarchive`, and `clear` subcommands to `fp issues`: `close` is a second terminal state that prevents a resolved issue from reopening on recurrence, `archive` / `unarchive` toggle a flag orthogonal to state, and `clear` bulk-resets a board with `--dry-run` preview and writes no suppression.

Preview which issues a board reset would remove before committing — useful after overhauling your agents and wanting a clean slate without silently suppressing recurring findings.

```
$ fp issues clear --dry-run
```

Permanently close a resolved issue so it stays closed even if the same audit finding recurs — use `close` instead of `resolve` for findings you have accepted as won't-fix.

```
$ fp issues close <issue-id>
```

— names each subcommand and its exact behavior v1.0.7

---

03 **Jev integration via FailproofAI Cloud** NEW 90

Adds Jev integration through FailproofAI Cloud via a new `machine` key preset (`events:add` + `policies:pull` + `jev:evaluate`), configured with `failproofai config --token <key>` ; it starts in shadow mode and switches to enforce mode with `failproofai jev setup --provider failproofai --mode enforce` .

— names the exact preset keys and enable commands v1.0.8

---

04 **Policy authority model gating what Jev can clear** NEW 85

Adds per-policy `authority` (`hard` or `reviewable`) and `reviewedBy` fields so only explicitly marked policies can be cleared by Jev; unmarked, invalid, or `alwaysOn` policies remain hard. Ships with 15 built-in reviewable policies covering workspace reads, env/secret exposure, destructive SQL, `rm -rf` , `kubectl` , `terraform` , cloud CLIs, and force-push, while `sudo` , `curl | sh` , and `block-work-on-main` remain hard. Jev consent clearing is limited to commands where every target was explicitly named by the user; clears are never granted for commands the scanner cannot fully parse (`$VAR` , $(...), backticks, heredocs, brace expansion, globs, `eval` , `sh -c`).

— names fields, policy list and parsing safeguards v1.0.8

---

05 **TypeScript telemetry SDK `@failproofai/sdk`** NEW 85

Adds `@failproofai/sdk` at `sdk/typescript` , supporting the same 15 events, wire format, spool directory, and Evaluator v2 protocol as the Python SDK, with instrument() adapters for `LangChain.js` / `LangGraph.js` , Vercel AI SDK, Mastra, and `LlamaIndex.TS` ; requires Node >= 20.9 and ships dual ESM+CommonJS with zero runtime dependencies.

— names package path, requirements and adapters v1.0.7

---

06 **semanticPolicies pack API for Jev checks** NEW 80

Adds semanticPolicies.add({ ... }) pack API so packs can ship Jev checks alongside regex policies; `failproofai publish` validates the Jev half, rejects reserved names, and

enforces `minCliVersion >= 1.0.8-beta.0`, writing it automatically. A pack that declares Jev checks requires `failproofai` 1.0.8 or later; older CLIs refuse the pack and name the remedy.

— names API, publish validation and version gate `v1.0.8`

---

07 **`human_input` event tracking in session collector** NEW

75

The session collector now emits a `human_input` event alongside `model_request`, distinguishing what a person actually typed from injected context (AGENTS.md, cron wrappers, sub-agent hand-offs) across Claude Code, Codex, OpenCode, pi, Hermes, and OpenClaw.

— names event, harnesses and injected-context sources `v1.0.7`

---

08 **Sandboxed server-authored evaluations** IMPROVED 75

Server-authored evaluations now run through a custom parser and interpreter instead of `eval` or `node:vm`, with every property read gated through a key-checking function, a `worker_threads` sandbox with V8 heap limits and a wall-clock terminate(), and hard refusal for any evaluation that cannot be sandboxed.

— mechanism detailed but no user-facing action `v1.0.7`

---

09 **New and expanded built-in regex policies** IMPROVED 70

Adds `warn-git-clean`, which warns before `git clean` deletes untracked or ignored files (`-d`, `-x`, `-X`), and extends `protect-env-vars` to catch whole-environment dumps via `set`, `export -p`, `declare -x`, and `os.environ` piped out.

— names exact policies and command patterns covered `v1.0.8`

---

10 **Jev observability panels in Cloud and local dashboards** NEW

65

Adds a Jev panel to the Cloud policy page showing a health strip (answered, fallbacks, latency, shadow and enforce machines), a Jev line on the decisions chart, and a cleared by Jev column per policy; adds a matching Jev panel to the local dashboard settings gear for write-only key entry.

— names UI elements but no deeper mechanism `v1.0.8`

---

11 **CLI/daemon protocol version enforcement** BREAKING 60

A CLI and daemon on different protocol versions now deny every tool call until they match; the fix is to run `failproofai update` immediately after `npm i -g failproofai@latest`.

— gives exact remedy command, mechanism otherwise thin `v1.0.8`

THINNER COVERAGE BELOW

12 **Cloud CLI first stable release (`fp-cloud-cli` 0.0.1)** NEW

50

Adds `fp-cloud-cli` 0.0.1, the first stable release of the Cloud CLI, on its own PyPI line.

— names package and version, no feature detail given v1.0.7

└──▷ **BREAKING ON UPGRADE**

! A CLI and daemon on different protocol versions deny every tool call until they match — run `failproofai update` immediately after `npm i -g failproofai@latest`.

! A pack that declares Jev checks requires `failproofai` 1.0.8 or later; older CLIs refuse the pack and name the remedy.

v1.0.8 NOTES ↗ LATEST STABLE

FailproofAI v1.0.8 adds Jev, a semantic classifier layer that sits above regex policies to cut false-positive blocks and catch more harmful agent tool calls.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.0.8
https://github.com/FailproofAI/failproofai.git
# already have the repo? check out this version:
$ git checkout v1.0.8
```

└──▷ **TRY IT**

Check Jev health and last-24-hour metrics — fallback rate, latency, and which policies it cleared — after enabling enforce mode.

```
$ failproofai jev status
```

Adds `failproofai jev setup --provider <typesafe|openrouter|vercel|cloudflare|custom>` (or `failproofai jev --url <url> --token <token>`) to configure Jev with a bring-your-own key from any supported provider; config is written to `~/.failproofai/jev.json` at mode 0600.

Adds `failproofai jev status` to display provider, mode, how many enabled policies Jev may clear, and the last 24 hours of evaluations, fallback rate, latency, and cleared calls.

Adds `failproofai jev test` to fire one real evaluation call and exit non-zero if the answer is too slow or wrong for hooks to use.

Adds `failproofai jev models` to list the models an endpoint serves and mark the one currently in use.

Adds `mode: off` option for any Jev provider that keeps the config in place but stops sending evaluations.

+9 more

└──▷ **BREAKING ON UPGRADE**

! A CLI and daemon on different protocol versions deny every tool call until they match — run `failproofai update` immediately after `npm i -g failproofai@latest`.

! A pack that declares Jev checks requires `failproofai` 1.0.8 or later; older CLIs refuse the pack and name the remedy.

v1.0.7 NOTES ↗ STABLE

FailproofAI v1.0.7 adds `human_input` event tracking, a TypeScript telemetry SDK, sandboxed evaluations, and four new `fp issues` subcommands.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.0.7
https://github.com/FailproofAI/failproofai.git
# already have the repo? check out this version:
$ git checkout v1.0.7
```

└──▷ **TRY IT**

Preview which issues a board reset would remove before committing — useful after overhauling your agents and wanting a clean slate without silently suppressing recurring findings.

```
$ fp issues clear --dry-run
```

Permanently close a resolved issue so it stays closed even if the same audit finding recurs — use `close` instead of `resolve` for findings you have accepted as won't-fix.

```
$ fp issues close <issue-id>
```

Adds `close`, `archive`, `unarchive`, and `clear` subcommands to `fp issues`: `close` is a second terminal state that prevents a resolved issue from reopening on

recurrence, `archive` / `unarchive` toggle a flag orthogonal to state, and `clear` bulk-resets a board with `--dry-run` preview and no suppression written.

The session collector now emits a `human_input` event alongside `model_request`, distinguishing what a person actually typed from injected context (AGENTS.md, cron wrappers, sub-agent hand-offs) across Claude Code, Codex, OpenCode, pi, Hermes, and OpenClaw.

Adds `@failproofai/sdk`, a TypeScript telemetry SDK at `sdk/typescript`, supporting the same 15 events, wire format, spool directory, and Evaluator v2 protocol as the Python SDK, with instrument() adapters for `LangChain.js` / `LangGraph.js`, Vercel AI SDK, Mastra, and `LlamaIndex.TS`; requires Node >= 20.9, ships dual ESM+CommonJS with zero runtime dependencies.

Server-authored evaluations now run through a custom parser and interpreter instead of `eval` or `node:vm`, with every property read gated through a key-checking function, a `worker_threads` sandbox with V8 heap limits and a wall-clock terminate(), and hard refusal for any evaluation that cannot be sandboxed.

Adds `fp-cloud-cli` 0.0.1, the first stable release of the Cloud CLI, on its own PyPI line.

WAS THIS USEFUL? 👍 👎

stride-gpt

Feedback ⓘ v0.20.0 NOTES ↗ RANK

Stride-GPT generates threat models and attack trees with large language models and the STRIDE framework.

> stride-gpt's v0.20.0 release adds OpenRouter as a model provider, checkpoint-based resumption for interrupted analysis runs, a verifiable-evidence threat reporting tool, paginated file reading, and per-phase token usage tracking.

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

01 **Paginated read_file by line range** IMPROVED 60

Pages `read_file` by line range with numbered output, allowing the agent to navigate large files without loading them whole.

— Names the function and its behaviour v0.20.0

THINNER COVERAGE BELOW

02 **Checkpoint-based resumption for analyze runs** NEW 55

Enables resuming an interrupted `/analyze` run from a checkpoint, avoiding full reruns after failures.

— Names the command and behaviour clearly v0.20.0

03 **report_threat tool with evidence and grace round** NEW 55

Introduces a `report_threat` tool that surfaces threats with verifiable evidence, plus a final grace round to catch late-breaking findings.

— Names the tool and its two mechanisms v0.20.0

04 **Per-phase and per-subsystem token tracking** NEW 45

Records token usage per phase and per subsystem, giving visibility into where LLM budget is spent across an analysis run.

— Describes what it does, no config surface named v0.20.0

05 **OpenRouter model provider support** NEW 35

Adds OpenRouter as a model provider, expanding the available LLM backends for threat analysis.

— Names the provider but no config detail v0.20.0

v0.20.0 NOTES ↗ LATEST STABLE

stride-gpt v0.20.0 adds OpenRouter as a model provider, checkpoint-based run resumption, and per-phase token tracking.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.20.0 https://github.com/mrwadams/stride-gpt.git
# already have the repo? check out this version:
$ git checkout v0.20.0
```

Adds OpenRouter as a model provider, expanding the available LLM backends for threat analysis.

Enables resuming an interrupted `/analyze` run from a checkpoint, avoiding full reruns after failures.

Introduces a `report_threat` tool that surfaces threats with verifiable evidence, plus a final grace round to catch late-breaking findings.

Pages `read_file` by line range with numbered output, allowing the agent to navigate large files without loading them whole.

Records token usage per phase and per subsystem, giving visibility into where LLM budget is spent across an analysis run.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ changelog · Sep 27, 2026 CHANGELOG ↗ RANK

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

> Anthropic shipped a new flagship model, on-demand context compaction, inline MCP tool definitions, and generally available cache diagnostics in this API update, alongside several breaking changes to existing behaviors.

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

01 Cache diagnostics reach general availability IMPROVED

85

Cache diagnostics are now GA; opting into diagnostics on every turn of a multi-turn session traces cache hit rates and links turns via `previous_message_id`. A request sending only the beta header without the `diagnostics` object no longer stores a fingerprint, so a later turn pointing `previous_message_id` at it reports `previous_message_not_found` — `diagnostics` must be included on every turn, with `"previous_message_id": null` on the first.

Opt into cache diagnostics on every turn of a multi-turn session to trace cache hit rates and link turns via previous_message_id.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 512,
    "diagnostics": {"previous_message_id": null},
    "messages": [{"role": "user", "content": "Start a new
analysis session."}]
  }'
```

— Names field, error code and behavioral requirement with example

changelog-20260927-27253774

└──▷ BREAKING ON UPGRADE

! On `claude-opus-5-5`, `thinking: {"type": "disabled", ...}` returns a 400 error — thinking cannot be disabled on this model.

! On the Claude API and Google Cloud, the earlier `computer_use` tool returns a 400 error on `claude-opus-5-5`; use the updated toolset instead (Amazon Bedrock continues to accept the old tool).

! For the cache-diagnostics extended thinking beta, a request that sends only the beta header (without the `diagnostics` object) no longer stores a fingerprint; a later turn pointing `previous_message_id` at it reports `previous_message_not_found`. Include `diagnostics` on every turn, with `"previous_message_id": null` on the first.

! The `filename` and `title` fields on Compliance API activity objects are now always empty or omitted, including for activities recorded before this change — any code reading these fields to get names will receive empty values.

changelog · Sep 27, 2026 CHANGELOG ↗ LATEST STABLE

Claude Opus 5.5, on-demand conversation compaction, inline tool definitions, and cache diagnostics GA ship in this Anthropic API update.

└──▷ **TRY IT**

Compact a long agentic conversation mid-session to stay within context limits without losing recent turns.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: compact-2026-09-04' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "compaction": {},
    "messages": [{"role": "user", "content": "Summarize our progress
so far."}]
  }'
```

Add a new tool mid-conversation without invalidating the prompt cache or rewriting the system prompt.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'anthropic-beta: inline-tools-2026-09-15' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 1024,
    "messages": [
      {"role": "user", "content": "Here is a new tool you can
use."},
      {"role": "user", "content": [{"type": "tool_addition", "tool":
{"type": "tool_definition", "definition": {"name": "run_query",
"description": "Run a SQL query", "input_schema": {"type": "object",
"properties": {"sql": {"type": "string"}}, "required":
["sql"]}}}}]},
      {"role": "user", "content": "Now use run_query to fetch the
top 10 users."}
    ]
  }'
```

Opt into cache diagnostics on every turn of a multi-turn session to trace cache hit rates and link turns via previous_message_id.

```
$ curl https://api.anthropic.com/v1/messages \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01' \
  -H 'content-type: application/json' \
  -d '{
    "model": "claude-opus-5-5",
    "max_tokens": 512,
    "diagnostics": {"previous_message_id": null},
    "messages": [{"role": "user", "content": "Start a new analysis
session."}]
  }'
```

Sends the `compact-2026-09-04` beta header and top-level `compaction` parameter to compact a conversation on demand; the API returns a signed `compaction` block to send in place of summarized messages on later requests.

With `inline-tools-2026-09-15` and the MCP connector's `mcp-client-2026-09-15` beta header, a tool definition can reference an MCP toolset; the response records each server's fetched tool list in an `mcp_tool_listing` block.

Introduces `claude-opus-5-5`, a new model for long-running agentic coding and knowledge work, at $4 / $20 USD per MTok, available on the Claude API.

└──▷ **BREAKING ON UPGRADE**

- ! On `claude-opus-5-5`, `thinking: {"type": "disabled", ...}` returns a 400 error — thinking cannot be disabled on this model.
- ! On the Claude API and Google Cloud, the earlier `computer_use` tool returns a 400 error on `claude-opus-5-5`; use the updated toolset instead (Amazon Bedrock continues to accept the old tool).
- ! For the cache-diagnostics extended thinking beta, a request that sends only the beta header (without the `diagnostics` object) no longer stores a fingerprint; a later turn pointing `previous_message_id` at it reports `previous_message_not_found`. Include `diagnostics` on every turn, with `"previous_message_id": null` on the first.
- ! The `filename` and `title` fields on Compliance API activity objects are now always empty or omitted, including for activities recorded before this change — any code reading these fields to get names will receive empty values.

WAS THIS USEFUL? 👍 👎

◆ **Open-Weight Models**

DeepSeek V4 Flash 0731 Feedback ⓘ

price cut · Sep 26, 2026 MODEL ↗

DeepSeek V4 Flash 0731 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4-flash-0731): $0.03 → $0.021 in, $0.32 → $0.32 out per million tokens (-30%), first seen 26 Sep 2026.

WAS THIS USEFUL? 👍 👎

NVIDIA Nemotron 3.5 Lightning Feedback ⓘ

price increase · Sep 26, 2026 MODEL ↗

Nemotron 3.5 Lightning is a chat model from NVIDIA.

Price increase on OpenRouter (nvidia/nemotron-3.5-lightning): $0.07 → $0.08 in, $0.20 → $0.20 out per million tokens (+14%), first seen 26 Sep 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.6 27B Feedback ⓘ

price increase · Sep 26, 2026 MODEL ↗

Qwen3.6 27B is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3.6-27b): $0.32 → $0.32 in, $2.70 → $3.20 out per million tokens (+19%), first seen 26 Sep 2026.

WAS THIS USEFUL? 👍 👎

OTHER / UNCATEGORIZED

◆ UNCATEGORIZED

Foremerge

Feedback ⓘ v0.5.0 NOTES ↗ RANK

Catch intent conflicts before code conflicts. The open-source coordination protocol for coding agents, built above Git.

> Foremerge v0.5.0 closes a command-injection path by locking the changeset validation endpoint to a trusted registry of checks and forcing validation to always run at the worktree root.

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

01 Trusted registry for changeset validation checks

BREAKING

93

`POST /v1/changesets/{id}/validate` now accepts only `{"check": "<name>"}` and runs solely checks registered via `foremerge checks set <name> [--timeout-seconds N] -- <argv...>`; a body containing `command`, `timeout_seconds`, `worktree`, or any other field is rejected with `INVALID_INPUT`, and an unregistered check name returns `NOT_FOUND`. This closes a command-injection path where callers could previously pass raw argument vectors. Migration: register each command with `foremerge checks set` and change request bodies to reference the registered name.

Register a test command in the trusted registry so it can be invoked safely via the HTTP API instead of passing raw argument vectors.

```
$ foremerge checks set run-unit-tests --timeout-seconds 120 --
cargo test --workspace
```

Trigger a registered check against a changeset over the HTTP API — replacing the old free-form body with the registry-scoped name.

```
$ curl -X POST
https://localhost:8080/v1/changesets/abc123/validate \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"check": "run-unit-tests"}'
```

— Exact endpoint, flag, error codes, and migration path given v0.5.0

02 Validation forced to worktree root

63

BREAKING

Validation now always runs at the root of the ChangeSet's worktree; passing a `--worktree` path naming a subdirectory on `foremerge changeset validate` or any other surface is rejected with `INVALID_INPUT`.

— Names flag and error code but no example given v0.5.0

└──▷ **BREAKING ON UPGRADE**

! `POST /v1/changesets/{id}/validate` no longer accepts a free-form request body (`command`, `timeout_seconds`, `worktree`, or any other field); the only valid body is `{"check": "<name>"}` where `<name>` is registered in the trusted registry. Callers that sent an argument vector will receive `INVALID_INPUT`. Migration: register each command with `foremerge checks set <name> [--timeout-seconds N] -- <argv...>` and update the request body to `{"check": "<name>"}`.

! Worktree paths naming a subdirectory are now rejected with `INVALID_INPUT` everywhere, including `foremerge changeset validate --worktree`; validation is forced to the worktree root.

v0.5.0 NOTES ↗ LATEST STABLE

Foremerge v0.5.0 locks `POST /v1/changesets/{id}/validate` to a trusted registry, closing a command-injection path.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.5.0 https://github.com/naw103/foremerge.git
# already have the repo? check out this version:
$ git checkout v0.5.0
```

└──▷ **TRY IT**

Register a test command in the trusted registry so it can be invoked safely via the HTTP API instead of passing raw argument vectors.

```
$ foremerge checks set run-unit-tests --timeout-seconds 120 -- cargo test --workspace
```

Trigger a registered check against a changeset over the HTTP API — replacing the old free-form body with the registry-scoped name.

```
$ curl -X POST https://localhost:8080/v1/changesets/abc123/validate
\
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"check": "run-unit-tests"}'
```

Changes `POST /v1/changesets/{id}/validate` to accept `{"check": "<name>"}` and run only checks registered in the repository's trusted registry via `foremerge checks set <name> [--timeout-seconds N] -- <argv...>`; bodies containing `command`, `timeout_seconds`, `worktree`, or any other field are rejected with `INVALID_INPUT`, and an unregistered name returns `NOT_FOUND`.

Validation now always runs at the root of the ChangeSet's worktree; any worktree path naming a subdirectory — via `--worktree` on `foremerge changeset validate` or any other surface — is rejected with `INVALID_INPUT`.

└──▷ BREAKING ON UPGRADE

- **!** `POST /v1/changesets/{id}/validate` no longer accepts a free-form request body (`command`, `timeout_seconds`, `worktree`, or any other field); the only valid body is `{"check": "<name>"}` where `<name>` is registered in the trusted registry. Callers that sent an argument vector will receive `INVALID_INPUT`. Migration: register each command with `foremerge checks set <name> [--timeout-seconds N] -- <argv...>` and update the request body to `{"check": "<name>"}`.
- **!** Worktree paths naming a subdirectory are now rejected with `INVALID_INPUT` everywhere, including `foremerge changeset validate --worktree`; validation is forced to the worktree root.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 27, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue