THE AI TOOLCHAIN

NO. 032

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 20, 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 101 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 17 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. verl cuts RL weight sync 10.3x by shipping only changed checkpoint shards to TorchTitan. Augment 's Cosmos can now run agent sandboxes on your own Kubernetes

vim ? help

cluster instead of vendor-hosted VMs. And xalgorix pauses scans on LLM rate limits and resumes them, so a long scan survives a quota wall instead of dying mid-run.

BUILD

Stop waiting on the weight-sync step between RL rollouts

The `delta_sharded` checkpoint format now works with the TorchTitan engine, transferring only changed shards rather than full weights, for a claimed 10.3x speedup. In RL post-training the actor must be refreshed after every policy update, so this is dead time on every iteration, not a one-off cost; a pluggable rollout router and a GPU-lending async trainer address the same idle-hardware problem from the scheduling side.

verl

DEPLOY

Run coding-agent sandboxes on your own cluster

Cosmos can now register a self-hosted Kubernetes cluster as a sandbox provider, installed from generated manifests or a CLI binary. For teams whose source can't leave their network, the execution environment for agent runs moves inside the perimeter without giving up the hosted control plane.

Augment Code

OPERATE

Let a long scan ride out a provider rate limit instead of failing

The Scanner Web UI gained caps on LLM request and subagent concurrency, plus automatic graceful pause-and-resume when rate limits, overload, or quota exhaustion hit. That removes the usual babysitting pattern — watch for 429s, kill the run, tune concurrency by hand, restart from scratch.

xalgorix

GOVERN

Get a signed record of what an agent actually did on the network

Agent Safe Pipeline pairs transparent network interception with provider-signed effect receipts (VP-3) and a published Agent-Safe Intent v1 spec with conformance vectors, so an agent's side effects become attestable rather than inferred from logs. LangSmith moves in the same direction on the trace side with `ls_provider` tagging for custom models in Gateway traces and a tightened permission requirement for dataset exports.

> agent-safe-pipeline · LangSmith

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.81 NOTES ↗ RANK

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

> xalgorix shipped new Scanner Web UI settings to cap LLM request and subagent concurrency, plus automatic graceful pause-and-resume of scans when LLM rate limits, overload, or quota exhaustion occur.

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

01 **Subagent concurrency control with FIFO queueing** NEW 80

Adds `XALGORIX_MAX_CONCURRENT_AGENTS` setting (category Runtime, default `3`, clamped `1..10`) in Scanner Web UI Settings; setting it to `1` runs specialist subagents serially via a deterministic FIFO queue without dropping specialist lanes or compromising evidence collection.

— Names config key plus concrete serial-execution mechanism and guarantee. v4.6.81

02 **LLM request concurrency cap setting** NEW 75

Adds `XALGORIX_LLM_MAX_INFLIGHT` setting (category `LLM`, default `4 × MaxInstances`, clamped `1..256`) in Scanner Web UI Settings to cap simultaneous outgoing LLM requests and prevent hitting provider rolling-window quota limits.

— Names exact config key, category, default and clamp range. v4.6.81

THINNER COVERAGE BELOW

03 **Graceful scan pause and resume on LLM limits** NEW 51

Scans now pause gracefully when cumulative LLM rate-limit or overload wait budgets expire, preserving findings, coverage, iteration counts, notes, and ledger state so the scan can resume; scans also pause immediately on quota exhaustion (insufficient balance/credits) without waiting for a budget timeout.

— Describes trigger conditions and preserved state but no config surface. v4.6.79

v4.6.81 NOTES ↗ LATEST STABLE

xalgorix v4.6.81 adds `XALGORIX_LLM_MAX_INFLIGHT` and `XALGORIX_MAX_CONCURRENT_AGENTS` to cap LLM request concurrency and control subagent execution.

└──▷ GET THIS VERSION

```
$ git clone --branch v4.6.81
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.81
```

Adds `XALGORIX_LLM_MAX_INFLIGHT` setting (category: `LLM`, default: `4 × MaxInstances`, clamped `1..256`) in Scanner Web UI Settings to cap simultaneous outgoing LLM requests and prevent hitting provider rolling-window quota limits.

Adds `XALGORIX_MAX_CONCURRENT_AGENTS` setting (category: `Runtime`, default: `3`, clamped `1..10`) in Scanner Web UI Settings; setting it to `1` runs specialist subagents serially via a deterministic FIFO queue without dropping specialist lanes or compromising evidence collection.

v4.6.79 NOTES ↗ STABLE

xalgorix v4.6.79 adds graceful scan pause-and-resume on LLM rate limits, quota exhaustion, and overload conditions.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v4.6.79
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.79
```

Pauses scans gracefully when cumulative LLM rate-limit or overload wait budgets expire, preserving findings, coverage, iteration counts, notes, and ledger state for resume.

Immediately pauses scans on quota exhaustion (insufficient balance/credits) without waiting for a budget timeout.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Letta Code

Feedback ⓘ v0.32.14 NOTES ↗ RANK

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

> Letta Code added Grok OAuth support for Cloud agents and changed how `/dream` and `/reflect` commands are routed based on reflection ownership.

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

01 **Grok OAuth for Cloud agents** NEW 65

Connects Grok OAuth to Cloud agents via the `connect_provider` protocol, which now accepts subscription OAuth tokens.

— Names protocol and token type but no setup steps v0.32.14

THINNER COVERAGE BELOW

02 **Reflection-ownership routing for /dream and /reflect** IMPROVED 30

Routes `/dream` and `/reflect` commands by reflection ownership.

— Only a brief mention, no mechanism or usage detail v0.32.14

v0.32.14 NOTES ↗ LATEST STABLE

Letta Code v0.32.14 adds Grok OAuth for Cloud agents and routes `/dream` and `/reflect` by reflection ownership.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.32.14 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.32.14
```

Connects Grok OAuth to Cloud agents via the `connect_provider` protocol, which now accepts subscription OAuth tokens.

WAS THIS USEFUL? 👍 👎

Earendil Works Pi

Feedback ⓘ v0.86.1 NOTES ↗ RANK

AI agent toolkit: unified LLM API, agent loop, TUI, coding agent CLI

> Pi's biggest additions this window are Meta Muse as a new LLM provider, a `/bug` command for structured crash and session reporting, and cost-aware prompt cache-warming; it also ships per-model compaction budgets, several new extension APIs, and breaking changes to provider stream and tool-call type contracts.

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

01 **/bug structured crash and session reporting** NEW 95

Adds `/bug [description]` to collect and submit structured bug reports, bundling redacted environment/model/provider/settings metadata plus either the raw session transcript or a model-written `summary.md`. Reports can be sent via Upload Report to `radius.pi.dev` (attributed to your Radius account if logged in) or written locally via Export as Zip; each submission is recorded as a `pi.bug-report` session entry with a UUID, and crashes are tracked in `~/.pi/agent/crashes.json`. The `PI_RADIUS_GATEWAY` environment variable overrides the Radius gateway origin used for these uploads and for Radius relay connections, enabling redirection to a self-hosted Radius deployment.

Report a bug after a failed session, attaching a model-written summary instead of the raw transcript to keep sensitive file contents out of the report.

```
$ /bug authentication stopped working after switching providers
```

Redirect bug report uploads to an internal self-hosted Radius instance instead of the public `radius.pi.dev` endpoint.

```
$ PI_RADIUS_GATEWAY=https://radius.internal.example.com pi
```

File a bug report mid-session after an unexpected failure, attaching a description so developers have context.

```
$ /bug agent looped indefinitely on a multi-file refactor and had to be interrupted
```

Report an unexpected model behaviour during a session, attaching a transcript summary for context.

```
$ /bug model produced truncated output after tool call
```

— Names command, config env var, output paths, and record format v0.86.0

02 **Prompt cache-warming to cut cache-miss costs** NEW 90

Adds `cacheWarming` config (value `"idle"`) that re-sends the last request with a one-token output budget before cache expiry to avoid full input re-pricing after inactivity, plus cost-aware warming during long tool runs. `PI_CACHE_RETENTION=long` declares the cache retention tier used by a request for accurate cache-lifetime scheduling, and a `promptCache` field in `models.json` declares prompt-cache lifetimes for custom models and non-Anthropic providers. Warming decisions are inspectable via `/session` (next decision, continuation probability, expected savings, threshold, estimated costs), extensions can override each decision through the `cache_warming_decision` event, and transcripts now show notices for significant cache misses, successful warming, compaction/branch-summary usage, and provider recovery diagnostics such as dropped Anthropic thinking blocks.

Enable idle cache warming to avoid paying full input price after long pauses between prompts — Pi will automatically refresh the cache before it expires.

```json
{"cacheWarming": "idle"}
```

— Names config key, env var, model field, and diagnostics surface v0.86.0

03 **Meta Muse provider support** NEW 81

Pi adds Meta (Muse subscription) as a supported LLM provider, authenticated via a `meta` device authorization flow triggered with `/login meta`, which auto-mints a Model API key that is re-minted approximately once per day. The key can also be supplied non-interactively via the `META_API_KEY` environment variable (e.g. in CI or a container without browser login) to access Muse Spark models.

Authenticate Pi with your Meta Muse subscription to mint a Model API key and start using Meta models.

```
$ /login meta
```

Authenticate with Meta Muse subscription to access Muse Spark models interactively.

```
$ /login meta
```

Authenticate with Meta Muse so Pi can access Muse Spark models in interactive sessions.

```
$ /login meta
```

Provide a Meta Muse API key non-interactively, e.g. in CI or a container where browser login is unavailable.

```
$ META_API_KEY=<your-muse-api-key> pi
```

— Names login command, env var, and refresh cadence v0.86.1

---

04 **Provider and tool-call type contract changes** BREAKING

80

Custom `pi-ai` providers must now read system prompts and tool declarations from `context.messages`, getCurrentSystemPrompt(), and getCurrentTools(), since stream inputs changed from `Context` to normalized `TranscriptContext` values. Additionally, `ToolCall.arguments` and `ToolResultMessage.details` are now restricted to JSON-compatible values, `ToolResultMessage` is now a conditional type, and `JsonValue` arrays are now readonly — custom providers and extensions using these types must be updated.

— Names exact types and methods but no migration example v0.86.0

---

05 **Per-model compaction budgets** NEW

80

Adds `compaction.modelOverrides` config with per-model `reserveTokens` and `keepRecentTokens` settings to control compaction budgets by model.

Override compaction token budgets for a large-context model to reserve more headroom and retain more recent tokens.

```yaml
compaction:
  modelOverrides:
    claude-opus-4-5:
      reserveTokens: 8000
      keepRecentTokens: 4000
```

Set per-model compaction budgets so a large-context model reserves more tokens before compaction kicks in.

```yaml
compaction:
  modelOverrides:
    claude-opus-4-5:
      reserveTokens: 8000
      keepRecentTokens: 4000
    claude-sonnet-4-5:
      reserveTokens: 4000
      keepRecentTokens: 2000
```

— Config key and runnable example given v0.86.0

06 **user_bash fails closed on errors** BREAKING 75

`user_bash` now fails closed: errors or invalid defined results abort the command without invoking later handlers or executing locally. Handlers must return `undefined` to continue propagation; otherwise they must return `{ operations }` or `{ result }`.

— Gives exact return contract needed to migrate handlers v0.86.0

07 **Override Anthropic fallback models** NEW 75

Adds `compat.allowedFallbackModels` configuration for overriding or disabling Anthropic server-side fallback models, letting users pin an explicit model and prevent silent substitution.

Prevent Anthropic server-side model fallbacks so Pi always uses your explicitly chosen model without silent substitution.

```yaml
compat:
  allowedFallbackModels: []
```

Override Anthropic server-side fallback models to pin a specific model and prevent silent substitution.

```yaml
compat:
  allowedFallbackModels:
    - claude-sonnet-4-5
```

— Config key with runnable YAML example v0.86.0

08 **Strict JSON-schema sampling by default** IMPROVED 65

Enables strict-prefer JSON-schema sampling by default for built-in tools without requiring `PI_EXPERIMENTAL`; extensions can opt out by re-registering with `constrainedSampling: false`.

— Names the opt-out flag but limited mechanism detail `product docs`

09 **before_agent_start extension hook** NEW 65

Adds `before_agent_start` extension hook support for transcript-backed mid-conversation system prompt and tool changes that survive resume and branch navigation while preserving cached prefixes.

— Describes mechanism and scope but no usage example `v0.86.0`

10 **Extension APIs for model calls and event handling** NEW 60

Adds ctx.modelRegistry.stream() and streamSimple() extension methods for making model calls through configured providers with resolved authentication, and adds an unsubscribe function returned from pi.on() so extensions can drop event handlers without restarting.

— Names two APIs but no usage example given `v0.86.0`

11 **Deferred tool loading for Fireworks Messages models** NEW 60

Adds native deferred tool loading for Fireworks Messages models via `ToolSearch` or `tool_search` as the loader name for prompt-prefix deferral.

— Names loader identifiers but no example or config path `v0.86.0`

THINNER COVERAGE BELOW

12 **Offline Radius model catalog** NEW 45

Adds the public Radius model catalog for immediate offline model selection, with cached and live gateway catalogs overlaid when available.

— Explains behaviour but no config or command named `v0.86.0`

13 **Click toggling for summary entries** IMPROVED 40

Adds click toggling for branch summaries, compaction summaries, and skill invocation entries.

— Names UI elements but not the navigation path `v0.86.0`

14 **Persistent compile cache for faster launches** IMPROVED 30

Enables Node's persistent compile cache before loading the bundled CLI runtime, reducing repeat launch time.

— No numbers, flag or config given, just prose v0.86.1

└──▷ **BREAKING ON UPGRADE**

! Inherited `pi-ai` provider stream inputs changed from `Context` to normalized `TranscriptContext` values; custom providers must now read system prompts and tool declarations from `context.messages`, getCurrentSystemPrompt(), and getCurrentTools().

! `ToolCall.arguments` and `ToolResultMessage.details` are now restricted to JSON-compatible values; `ToolResultMessage` is now a conditional type; and `JsonValue` arrays are now readonly — custom providers and extensions using these types must be updated.

! `user_bash` now fails closed: errors or invalid defined results abort the command without invoking later handlers or executing locally. Return `undefined` to continue propagation; otherwise return `{ operations }` or `{ result }`.

! Custom pi-ai providers must now read system prompts and tool declarations from `context.messages` using getCurrentSystemPrompt() and getCurrentTools(), as stream inputs changed from `Context` to normalized `TranscriptContext` values.

! `ToolCall.arguments` and `ToolResultMessage.details` are now restricted to JSON-compatible values; `ToolResultMessage` is now a conditional type; `JsonValue` arrays are now readonly.

v0.86.1 NOTES ↗ LATEST STABLE

Pi v0.86.1 adds Meta Muse provider support via `/login meta` or `META_API_KEY`, plus faster repeat launches.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.86.1 https://github.com/earendil-
works/pi.git
# already have the repo? check out this version:
$ git checkout v0.86.1
```

└──▷ **TRY IT**

Authenticate with Meta Muse so Pi can access Muse Spark models in interactive sessions.

```
$ /login meta
```

Provide a Meta Muse API key non-interactively, e.g. in CI or a container where browser login is unavailable.

```
$ META_API_KEY=<your-muse-api-key> pi
```

Adds Meta Muse provider access via `/login meta` with automatic Model API key refresh, or by setting `META_API_KEY` to use Muse Spark models.

Enables Node's persistent compile cache before loading the bundled CLI runtime, reducing repeat launch time.

v0.86.0 NOTES ↗ STABLE

Pi v0.86.0 adds cache warming, `/bug` reporting, offline Radius catalog, per-model compaction budgets, and new extension APIs.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.86.0 https://github.com/earendil-works/pi.git
# already have the repo? check out this version:
$ git checkout v0.86.0
```

└──▷ **USE IT**

Override compaction token budgets for a large-context model to reserve more headroom and retain more recent tokens.

```yaml
compaction:
  modelOverrides:
    claude-opus-4-5:
      reserveTokens: 8000
      keepRecentTokens: 4000
```

Prevent Anthropic server-side model fallbacks so Pi always uses your explicitly chosen model without silent substitution.

```yaml
compat:
  allowedFallbackModels: []
```

File a bug report mid-session after an unexpected failure, attaching a description so developers have context.

```
$ /bug agent looped indefinitely on a multi-file refactor and had to be interrupted
```

Adds `compaction.modelOverrides` config with per-model `reserveTokens` and `keepRecentTokens` settings to control compaction budgets by model.

Adds `compat.allowedFallbackModels` configuration for overriding or disabling Anthropic server-side fallback models.

Adds ctx.modelRegistry.stream() and streamSimple() extension methods for making model calls through configured providers with resolved authentication.

Adds an unsubscribe function from pi.on() so extensions can drop event handlers without restarting.

Adds native deferred tool loading for Fireworks Messages models via `ToolSearch` or `tool_search` as the loader name for prompt-prefix deferral.

+5 more

└─▷ **BREAKING ON UPGRADE**

! Custom pi-ai providers must now read system prompts and tool declarations from `context.messages` using getCurrentSystemPrompt() and getCurrentTools(), as stream inputs changed from `Context` to normalized `TranscriptContext` values.

! `ToolCall.arguments` and `ToolResultMessage.details` are now restricted to JSON-compatible values; `ToolResultMessage` is now a conditional type; `JsonValue` arrays are now readonly.

! `user_bash` now fails closed: errors or invalid defined results abort the command without invoking later handlers or executing locally. Return `undefined` to continue propagation; otherwise return `{ operations }` or `{ result }`.

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.6.0 NOTES ↗

RANK

Zeroshot orchestrates executor-verifier agent loops that implement and validate software changes while logging workflow state to SQLite.

> Zeroshot's biggest addition this window is a standalone browser UI and native Windows support, alongside new `zeroshot resume` and `zeroshot update` commands, explicit PR delivery modes, and experimental ACP agent exposure for persistent sessions.

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

01 `zeroshot update` command for binary and skills NEW

90

Adds a `zeroshot update` command to refresh the Zeroshot executable and unmodified managed skill copies from a checksummed GitHub Release asset, preserving user-edited or conflicting skills and reporting their exact paths. The command verifies bounded downloads, the checksum manifest, archive contents, and the staged binary before replacing the installed executable.

Refresh the Zeroshot binary and managed skill copies to the latest release without overwriting any skills you have edited.

```
$ zeroshot update
```

Safely upgrade the installed Zeroshot binary with checksum and archive verification before replacing the executable.

```
$ zeroshot update
```

— Exact command with verification steps and safety behaviour v10.4.0 v10.5.0

02 Browser UI via `zeroshot ui` NEW

85

Adds a `zeroshot ui` command to edit profiles, monitor active runs, inspect transcripts, and replay recorded history in the browser. The embedded UI is served at `/ui/` on any direct target with no Node server required.

Launch the browser UI to monitor active runs and inspect transcripts without spinning up a separate Node server.

```
$ zeroshot ui
```

— Names exact command and serve path; full behaviour described v10.4.0

03 **Native Windows support** NEW 75

Adds native Windows support for local and detached runs, observation, cancellation, Git delivery, and the embedded UI, using private named pipes, protected state, and Job Objects for transport and process cleanup.

— Mechanism named but no runnable command given v10.4.0

04 **`zeroshot resume` for failed runs** NEW 75

Adds a `zeroshot resume` command to continue failed local or Docker runs by reusing the retained dirty workspace while preserving prior results and logs.

Resume a failed run after a crash or interruption, picking up retained workspace state instead of starting from scratch.

```
$ zeroshot resume
```

— Exact command with clear before/after behaviour v10.5.0

05 **Deterministic GitHub Release notes generation** NEW 65

Generates deterministic GitHub Release notes from first-parent squash commits since the preceding canonical tag, passing one immutable notes artifact from release planning through to publication.

— Mechanism explained but no command to invoke it v10.5.0

06 **npm skill installs for Codex, Copilot, Claude Code** IMPROVED 65

The npm package now installs the Zeroshot skill for Codex, GitHub Copilot, and Claude Code, preserves edited copies, and reports incomplete installations instead of overwriting them.

— Names three targets but no install command given v10.4.0

07 **Explicit delivery modes for graph checks** NEW 60

Adds explicit delivery modes — local-only, push, PR-readiness, and merge — giving practitioners fine-grained control over how changes are delivered after graph checks pass.

— Modes named but no flag or command shown v10.5.0

THINNER COVERAGE BELOW

08 **Experimental ACP agent exposure over stdio** NEW 55

Exposes saved local Zeroshot profiles as experimental ACP agents over stdio, keeping workspace and node-provider sessions alive across prompts while each prompt remains a separate durable run with fresh Git provenance.

— Marked experimental with no CLI entry point shown v10.6.0

09 **Autonomous PR feedback and delivery** NEW 45

Makes PR feedback actionable by default, with runtime and CLI opt-outs available, and supports autonomous PR delivery while allowing missing human approval.

— Behaviour change described without named flags v10.5.0

v10.6.0 NOTES ↗ LATEST STABLE

Zeroshot v10.6.0 exposes local profiles as experimental ACP agents over stdio for persistent, observable sessions.

└──▷ GET THIS VERSION

```
$ git clone --branch v10.6.0 https://github.com/the-open-engine/zeroshot.git
# already have the repo? check out this version:
$ git checkout v10.6.0
```

v10.5.0 NOTES ↗ STABLE

Zeroshot v10.5.0 adds `zeroshot resume` for failed runs, autonomous PR delivery modes, and self-updating managed skills via `zeroshot update`.

└──▷ GET THIS VERSION

```
$ git clone --branch v10.5.0 https://github.com/the-open-engine/zeroshot.git
# already have the repo? check out this version:
$ git checkout v10.5.0
```

└──▷ TRY IT

Resume a failed run after a crash or interruption, picking up retained workspace state instead of starting from scratch.

```
$ zeroshot resume
```

Refresh the Zeroshot binary and managed skill copies to the latest release without overwriting any skills you have edited.

```
$ zeroshot update
```

Adds `zeroshot resume` command to continue failed local or Docker runs by reusing the retained dirty workspace while preserving prior results and logs.

Adds `zeroshot update` command to refresh the Zeroshot executable and unmodified managed skill copies from a checksummed GitHub Release asset, while preserving user-edited or conflicting skills and reporting their exact paths.

Adds explicit delivery modes — local-only, push, PR-readiness, and merge — giving practitioners fine-grained control over how changes are delivered after graph checks pass.

Makes PR feedback actionable by default, with runtime and CLI opt-outs available, and supports autonomous PR delivery while allowing missing human approval.

Generates deterministic GitHub Release notes from first-parent squash commits since the preceding canonical tag, passing one immutable notes artifact from release planning through to publication.

v10.4.0 NOTES ↗ STABLE

Zeroshot v10.4.0 adds a standalone browser UI, native Windows support, and a `zeroshot update` command with verified binary delivery.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v10.4.0 https://github.com/the-open-engine/zeroshot.git
# already have the repo? check out this version:
$ git checkout v10.4.0
```

└──▷ **TRY IT**

Launch the browser UI to monitor active runs and inspect transcripts without spinning up a separate Node server.

```
$ zeroshot ui
```

Safely upgrade the installed Zeroshot binary with checksum and archive verification before replacing the executable.

```
$ zeroshot update
```

Adds `zeroshot ui` command to edit profiles, monitor active runs, inspect transcripts, and replay recorded history in the browser.

Serves the embedded UI at `/ui/` on any direct target — no Node server required.

Adds `zeroshot update` command that verifies bounded downloads, the checksum manifest, archive contents, and the staged binary before replacing the installed executable.

Adds native Windows support for local and detached runs, observation, cancellation, Git delivery, and the embedded UI — using private named pipes, protected state, and Job Objects for transport and process cleanup.

npm package now installs the Zeroshot skill for Codex, GitHub Copilot, and Claude Code, preserves edited copies, and reports incomplete installations instead of overwriting them.

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-08-21 CODE ↗

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

Diagram Design: Adds `ridgeline` as a `Line` variant in `type-line.md`, with a dedicated `verify-ridgeline.py` gate that derives a single shared amplitude to catch per-ridge normalisation lies.

└──▷ **GET THIS VERSION**

```
$ git clone --branch commits-2026-08-21
https://github.com/cathrynlavery/diagram-design.git
# already have the repo? check out this version:
$ git checkout commits-2026-08-21
```

└──▷ **TRY IT**

Verify a marimekko chart's column widths and segment heights match their declared values before publishing.

```
$ python3 scripts/verify-marimekko.py path/to/example-marimekko.html
```

Verify a bubble chart's area encoding, axis scales, and paint order are geometrically honest.

```
$ python3 scripts/verify-bubble.py path/to/example-bubble.html
```

Verify a ridgeline chart uses a single shared amplitude so no individual ridge is silently rescaled.

```
$ python3 scripts/verify-ridgeline.py path/to/example-ridgeline.html
```

Adds `bubble` as a `Scatter` variant in `type-scatter.md` using `data-size`, `data-name`, and `data-tick` bindings, with a `verify-bubble.py` gate covering axis scales, area-proportional encoding, and paint order.

Adds `marimekko` as a `Treemap` variant with a `verify-marimekko.py` gate enforcing nine geometric invariants (column-width-to-category-share, segment tiling, plot alignment, etc.).

WAS THIS USEFUL?  

Superset

Feedback ⓘ desktop-v1.30.0 NOTES ↗ RANK

Superset is an agentic IDE to orchestrate 100+ coding agents in parallel. Run any agent with your own subscription.

> Superset's v1.30 release adds a Slack agent integration with thread-based sessions and plugin bridging, overhauls cloud workspaces into a v2 bundle model, and replaces browser PRs/pages with native desktop panes alongside new OAuth plugins and mobile navigation improvements.

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

01 **Slack agent integration for running turns** NEW 85

Introduces a Slack agent foundation with a shared handler, execution guard, thread window, and run-budget enforcement, plus a `!stop` command to end a running turn. Adds Slack thread sessions so replies in a thread reach the agent without requiring an @mention, queues Slack replies that arrive mid-turn and hands them back to the agent after the current turn completes, and bridges the Slack agent to the user's Linear and GitHub plugins.

Stop a running Slack agent turn mid-execution by posting `!stop` in the Slack thread.

```
$ !stop
```

— Details mechanism and includes the named stop command desktop-v1.30.0

02 **Local and cloud workspace mode controls** NEW 78

Adds `--local` / `--host` flags to the CLI, MCP, SDK, and cloud commands so workspaces default to running in the cloud, with an opt-in to a local or host workspace; adds environment-only workspace creation and supports explicit local workspaces on shared checkouts.

Force a workspace to run on a local host instead of the cloud default.

```
$ superset workspace create --local
```

— Names exact flags and includes a runnable command desktop-v1.30.0

03 **Native browser and page panes with public publishing** NEW 62

Replaces browser PRs and pages with native panes in the desktop app, adds viewport pinch zoom to browser and page panes, supports publishing pages without a workspace, adds public pages with thumbnails in link unfurls, auto-opens a newly created page in the browser from CLI and plugins, and adds author filtering for pages in the desktop app.

— Lists concrete UI capabilities but no exact commands `desktop-v1.30.0`

THINNER COVERAGE BELOW

04 **Cloud Workspaces v2** NEW 57

Introduces Cloud Workspaces v2, treating the workspace as a bundle with multi-repository environments and brokered credentials. Cloud workspaces now operate on their own branch rather than the base branch, and support attaching images to a cloud workspace at creation time in both cloud and desktop.

— Describes mechanism but no exact command or config key `desktop-v1.30.0`

05 **Mobile workspace and navigation improvements** IMPROVED 55

Mobile app now keeps up to four terminal sessions live per workspace screen, enables No project session creation from the composer and Home, points an offline host at Remote Access setup, and adds a header org button that navigates directly to Settings.

— Names concrete mobile UI changes and a session limit `desktop-v1.30.0`

06 **OAuth plugin integrations for Notion, Slack, Gmail** NEW 35

Adds dynamic OAuth clients plus new Notion, Slack, and Gmail plugin integrations.

— Names integrations but gives no config or usage detail `desktop-v1.30.0`

07 **Session account usage shown in pane headers** NEW 21

Shows session account usage in pane headers.

— Bare description with no mechanism or scope `desktop-v1.30.0`

`desktop-v1.30.0` NOTES ↗ LATEST STABLE

Superset v1.30 adds Slack agent integration, public pages, cloud workspaces v2, native browser/page panes, and Notion/Gmail/Slack OAuth plugins.

└──▷ GET THIS VERSION

```
$ git clone --branch desktop-v1.30.0 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout desktop-v1.30.0
```

└──▷ **TRY IT**

Force a workspace to run on a local host instead of the cloud default.

```
$ superset workspace create --local
```

Stop a running Slack agent turn mid-execution by posting `!stop` in the Slack thread.

```
$ !stop
```

Adds `--local` / `--host` flags to CLI, MCP, SDK, and cloud commands so workspaces default to the cloud, with opt-in to a local/host workspace; adds environment-only workspace creation.

Introduces a Slack agent foundation with a shared handler, execution guard, thread window, run-budget enforcement, and `!stop` command to end a running turn.

Adds Slack thread sessions so replies in a thread reach the agent without requiring a @mention.

Queues Slack replies that arrive mid-turn and hands them back to the agent after the current turn completes.

Bridges the Slack agent to the user's Linear and GitHub plugins.

+16 more

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

verl

Feedback ⓘ v0.9.1 NOTES ↗ RANK

verl is a reinforcement learning post-training framework for language models, implementing the HybridFlow programming model over Megatron and FSDP backends.

> verl v0.9.1 adds a pluggable rollout router, extends the delta_sharded checkpoint format to the TorchTitan engine for a 10.3x weight-sync speedup, introduces a GPU-lending async trainer and a post-checkpoint callback hook, and moves dependency management to uv while removing the mindspeedllm backend and several legacy config keys.

└──▷ **WHAT SHIPPED · 20 FEATURES** most completely described first ? what's the number?

01 Pluggable rollout router NEW

95

Adds `rollout.router_config_path` config key to load an external load-balancer plugin from YAML, specified by a `router_class` fully-qualified name with Hydra `defaults` composition and `pkg://` URIs. The built-in sticky-session balancer moves into `verl/workers/rollout/router.py` behind a `RequestLoadBalancer` protocol and a get_router_handle() factory, letting users swap in custom request-routing logic.

Load an external rollout load-balancer plugin, replacing the built-in sticky-session balancer, for custom request routing logic.

yaml

```yaml
rollout:
  router_config_path: /path/to/my_router.yaml
  # my_router.yaml contains:
  # defaults:
  #   - _self_
  # router_class: myproject.routing.MyLoadBalancer
```

— Names config key, protocol and factory, with runnable example. v0.9.1

02 torch_memory profiler with OOM observer NEW

90

Adds a `torch_memory` profiler with a process-local OOM observer that logs allocation size, call stack, and allocator summary and dumps a snapshot under `<save_path>/oom_<timestamp>/` ; `memory_snapshot_num_steps` retains memory history across profiled steps.

— Names profiler, config key and exact dump path. v0.9.1

03 Checkpoint save callback hook NEW

88

Adds `trainer.checkpoint_callback_class` config key naming a user-defined `CheckpointCallback` whose `on_save` hook fires after every checkpoint save, an extension point for uploads, model registries, and retention policies without forking the trainer.

Register a post-checkpoint hook to push saved checkpoints to a model registry without forking the trainer.

```yaml
trainer:
  checkpoint_callback_class:
myproject.callbacks.UploadCheckpointCallback
```

— Named config key and hook with a runnable example. v0.9.1

04 **GPU-lending async trainer** NEW 85

Adds `trainer.v1.separate_async.enable_switch` (off by default) to lend idle trainer GPUs to generation during async rollout steps, driven by an adaptive starvation threshold.

Enable idle trainer GPUs to assist generation during async rollout to reduce GPU starvation.

```yaml
trainer:
  v1:
    separate_async:
      enable_switch: true
```

— Exact flag, default, and runnable config example. v0.9.1

05 **Fully-async V0 semantics for V1 trainers** IMPROVED 85

Adds V0-style fully-async semantics to V1 async trainers via `actor_rollout_ref.hybrid_engine=False` (rollout served exclusively by the standalone pool), fractional `num_warmup_batches` (e.g. `1.5`), and `max_off_policy_threshold=null` to disable off-policy version control entirely; defaults preserve prior behavior.

— Names three exact flags and their semantics, no full example. v0.9.1

06 **DAPO group filtering** BREAKING 83

Adds DAPO group filtering support via `algorithm.filter_groups.enable=True`, reading the canonical pre-KL reward from `rm_scores`;

`algorithm.filter_groups.metric` now defaults to `reward`, so DAPO runs that previously omitted this key may see changed group-filtering behavior.

— Names config keys, source field and the default-change impact. v0.9.1

07 **delta_sharded checkpoint sync for TorchTitan** IMPROVED

80

Extends the `delta_sharded` checkpoint format to the TorchTitan engine across FSDP, FSDP+TP, FSDP+EP, HSDP, HSDP+TP, and HSDP+EP sharding modes, achieving 10.3x faster weight sync at Qwen3-8B on 32 GPUs (3.34s vs 34.26s).

— Named format, engine and modes with concrete benchmark, no command. v0.9.1

08 **Liger Kernel fused cross-entropy for actor updates** IMPROVED

80

When Liger Kernel v0.8.2 is installed, actor updates use `LigerFusedLinearScaledCrossEntropyFunction`, delivering 13.5% faster actor updates and 5.6% lower actor peak memory (0.02803 -> 0.02424 ms/token, 35.68 -> 33.70 GiB, measured across four H100 seeds).

— Concrete benchmark and named function, activation is implicit not a flag. v0.9.1

09 **VL continuous token builder for multimodal RL** BREAKING

73

Adds `VLContinuousTokenBuilder`, extending the Continuous Token path to `SingleTurnAgentLoop` and `ToolAgentLoop` for multimodal RL training. The builder is now composed from `VLContinuousTokenMixin` plus a text-family builder, and the agent loop postprocesses multimodal tensors over the full message history, a breaking change to the prior `ContinuousTokenBuilder` surface.

— Names builder classes and loops plus the breaking surface change. v0.9.1

10 **uv-based dependency management** BREAKING

70

Moves dependency management to `pyproject.toml` and a committed `uv.lock` covering vLLM, SGLang x FSDP, and Megatron extras, with aarch64 support; `uv` is now the supported installation path for verl and its backend extras.

— Names lockfile, covered backends and the new install path. v0.9.1

11 **multi_sender mode for NCCL checkpoint engine** NEW

65

Adds an opt-in `multi_sender` mode to the NCCL checkpoint engine that admits actor rank 0's node-local NVLink peers as broadcast relays, letting weight sync drive more than one NIC per node.

— Named mode and mechanism but no exact config key or example. v0.9.1

12 **Profiler post-hook and result relocation** NEW 60

Adds `profiler.finish_hook_cmd` and `finish_hook_ranks` config keys for a post-hook on the torch profiler, plus `global_profiler.relocate_results` for relocating profiler output.

— Names three config keys but no usage detail beyond names. v0.9.1

13 **Removed legacy backend and config keys** DEPRECATED 60

Removes the `mindspeedllm` backend engine and drops config keys Megatron `grad_offload`, `actor.router_replay` (replaced by the engine config), and the unused ref router replay config.

— Names removed backend and config keys but gives no migration detail. v0.9.1

THINNER COVERAGE BELOW

14 **async_activation_offload for VeOmni training** NEW 45

Adds `async_activation_offload` config key for VeOmni training.

— Names config key but no mechanism or example. v0.9.1

15 **Prefix-cache hit counts in vLLM rollout** NEW 45

Surfaces prefix-cache hit counts in `TokenOutput` for vLLM rollout.

— Names the field but no usage or downstream effect. v0.9.1

16 **New fsdpturbo backend engine** NEW 35

Adds a new `fsdpturbo` backend engine.

— Bare name of new backend, no mechanism or usage given. v0.9.1

17 **transformers pinned to 5.9.0** BREAKING 35

Upgrades and pins transformers to `5.9.0` across CI and images.

— Version named but no rationale or migration detail. v0.9.1

18 **Video features forwarded to SGLang rollout** NEW 30

Forwards video features to SGLang rollout.

— One-line description with no mechanism or scope. v0.9.1

19 **Agent Loop telemetry to RL-Insight** NEW 30

Adds Agent Loop telemetry forwarding to RL-Insight.

— One-line description with no fields or configuration named. v0.9.1

20 **Slurm Ray network interface configurability** IMPROVED

Adds configurability for the Ray network interface when running under Slurm.

— No config key, flag, or mechanism named. v0.9.1

└──▷ BREAKING ON UPGRADE

! Dependency management moved to uv: environments are now described by `pyproject.toml` and a committed `uv.lock`; `uv` is the supported installation path for verl and its backend extras.

! `ContinuousTokenBuilder` surface changed: VL builders are now composed from `VLContinuousTokenMixin` plus a text-family builder, and the agent loop postprocesses multimodal tensors over the full message history.

! `actor_rollout_ref.hybrid_engine=False` is now honored under `separate_async` (rollout served exclusively by the standalone pool); `num_warmup_batches` now accepts fractional values; `max_off_policy_threshold` now accepts `null`. Defaults preserve the previous behavior.

! `algorithm.filter_groups.metric` now defaults to `reward` and resolves against the canonical pre-KL `rm_scores` — DAPO runs that previously omitted this key may see changed group-filtering behavior.

! Removed config keys: Megatron `grad_offload`, `actor.router_replay` (replaced by the engine config), and the unused ref router replay config.

! The `mindspeedllm` backend engine was removed.

! transformers is pinned to 5.9.0.

v0.9.1 NOTES ↗ LATEST STABLE

verl v0.9.1 adds pluggable rollout router, delta_sharded vLLM/TorchTitan weight sync, GPU-lending async trainer, and uv-based dependency management.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.9.1 https://github.com/verl-project/verl.git
# already have the repo? check out this version:
$ git checkout v0.9.1
```

└──▷ USE IT

Register a post-checkpoint hook to push saved checkpoints to a model registry without forking the trainer.

yaml

```yaml
trainer:
  checkpoint_callback_class:
myproject.callbacks.UploadCheckpointCallback
```

Enable idle trainer GPUs to assist generation during async rollout to reduce GPU starvation.

```yaml
trainer:
  v1:
    separate_async:
      enable_switch: true
```

Load an external rollout load-balancer plugin, replacing the built-in sticky-session balancer, for custom request routing logic.

```yaml
rollout:
  router_config_path: /path/to/my_router.yaml
  # my_router.yaml contains:
  # defaults:
  #   - _self_
  # router_class: myproject.routing.MyLoadBalancer
```

Adds `rollout.router_config_path` config key to load an external router plugin from YAML (by `router_class` FQN with Hydra `defaults` composition and `pkg://` URIs), moving the load balancer into `verl/workers/rollout/router.py` behind a `RequestLoadBalancer` protocol and get_router_handle() factory.

Adds `trainer.v1.separate_async.enable_switch` config key (off by default) to lend idle trainer GPUs to generation during async rollout steps, driven by an adaptive starvation threshold.

Adds `trainer.checkpoint_callback_class` config key naming a user-defined `CheckpointCallback` whose `on_save` hook fires after every checkpoint save — an extension point for uploads, model registries, and retention policies without forking the trainer.

Extends `delta_sharded` checkpoint format to the TorchTitan engine (FSDP, FSDP+TP, FSDP+EP, HSDP, HSDP+TP, HSDP+EP), achieving 10.3x faster weight sync at Qwen3-8B on 32 GPUs (3.34 s vs 34.26 s).

Adds opt-in `multi_sender` mode to the NCCL checkpoint engine, admitting actor rank 0's node-local NVLink peers as broadcast relays to drive more than one NIC per node.

+14 more

└──▷ **BREAKING ON UPGRADE**

! Dependency management moved to uv: environments are now described by `pyproject.toml` and a committed `uv.lock`; `uv` is the supported installation

path for verl and its backend extras.

- ! `ContinuousTokenBuilder` surface changed: VL builders are now composed from `VLContinuousTokenMixin` plus a text-family builder, and the agent loop postprocesses multimodal tensors over the full message history.
- ! `actor_rollout_ref.hybrid_engine=False` is now honored under `separate_async` (rollout served exclusively by the standalone pool); `num_warmup_batches` now accepts fractional values; `max_off_policy_threshold` now accepts `null`. Defaults preserve the previous behavior.
- ! `algorithm.filter_groups.metric` now defaults to `reward` and resolves against the canonical pre-KL `rm_scores` — DAPO runs that previously omitted this key may see changed group-filtering behavior.
- ! Removed config keys: Megatron `grad_offload`, `actor.router_replay` (replaced by the engine config), and the unused ref router replay config.
- ! The `mindspeedllm` backend engine was removed.
- ! transformers is pinned to 5.9.0.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.53 NOTES ↗ RANK

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

> HyperFrames' biggest addition this window is resumable segmented capture for long-form renders, backed by new `--resume` / `--keep-segments` flags, an `HF_SEGMENTED_CAPTURE` env var and a content-addressed segment manifest, alongside a new `hyperframes timeline` CLI subcommand, composition linting in Studio, and expanded host-app integration APIs and dockable panel layouts.

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

01 Segmented capture for long-form renders NEW

95

Adds `--resume` and `--keep-segments` flags to the render command for segmented capture, allowing interrupted long renders to be continued. The `HF_SEGMENTED_CAPTURE` environment variable routes long renders to segmented capture without changing flags, a segment manifest with a content-addressed plan hash makes long-form renders resumable, and multi-worker parallel segmented capture now runs off a shared queue.

Resume an interrupted long-form render without re-capturing already-completed segments, preserving disk-written frames.

```
$ hyperframes render --resume --keep-segments
```

Route long renders through segmented capture in CI without changing render command flags — set once in the environment.

```
$ HF_SEGMENTED_CAPTURE=1 hyperframes render
```

— Names exact flags, env var, and runnable commands v0.8.51

02 Render event metadata and Chrome memory sampling IMPROVED

73

Render events now carry the browser's build and install-path shape, giving CLI consumers richer metadata per render, and also emit Chrome memory, capture path, and segment fields. HyperFrames adds Chrome process memory sampling per capture session via a zero-dependency per-pid RSS sampler (`ps` / `tasklist`).

— Names sampler mechanism and event fields, no direct command v0.8.52 v0.8.51

03 Host-app integration surfaces for Studio NEW

73

Exports a stable `styles.css` subpath, header logo, and button sizes from Studio for host-app integration. Host apps can also place overlays on the preview and restyle its stage, and can import the render queue directly from the package.

— Names concrete surfaces exported for host apps v0.8.53 v0.8.52

04 **`hyperframes timeline` CLI subcommand** NEW 70

Adds `hyperframes timeline` to print a project's tracks and clips, letting users inspect timeline structure from the command line before rendering.

Inspect all tracks and clips in a project before rendering to verify timeline structure from the command line.

```
$ hyperframes timeline
```

— Runnable command with usage example v0.8.51

05 **`streamingEncodeDurationCapEnabled` engine config flag** NEW 70

Adds `streamingEncodeDurationCapEnabled` engine config flag (default off) to gate streaming encode duration capping.

— Named config flag with default value v0.8.51

06 **Composition linting in Studio** NEW 65

Adds linting for nested structure, missing timing, and caption tracks on the root composition, and detects where the gsap timeline disagrees with timing attributes in the Parsers layer. The Lint badge now pulses in Studio while an error finding exists.

— Names checks and UI indicator but no exact path v0.8.51

07 **Studio panel layout and accessibility improvements** IMPROVED 62

Moves all Studio panels onto a dockable, persisted layout that remembers arrangement between sessions, and adds the Renders panel to the shared primitives. Dock dividers and tabs are now keyboard-navigable, side columns fit narrow windows, and undo/redo have moved into the timeline toolbar.

— UI navigation described but no exact command v0.8.53 v0.8.52

THINNER COVERAGE BELOW

08 **Speed ramp rendering and automation** NEW 57

Renders speed ramps with the same source-time mapping as the preview, and supports speed ramps as a rate lane in `data-automation`.

— Names data-automation lane but no runnable step v0.8.51

09 **3D-motion registry blocks in Catalog** NEW 45

Adds seven 3D-motion pieces as installable registry blocks to the Catalog.

— Names Catalog and count but no install steps v0.8.52

10 **Composition duration derived from clips** IMPROVED 38

Compositions with no authored length now derive their duration from their clips.

— Behavior described but no named surface v0.8.51

11 **PR check for reversed before/after captures** NEW 35

Scripts now fail a PR whose After capture is a Before capture.

— Bare description of a new check v0.8.52

v0.8.53 NOTES ↗ LATEST STABLE

HyperFrames v0.8.53 adds a dockable persisted panel layout and exposes a stable `styles.css` subpath for host-app integration.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.53 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.53
```

Exports a stable `styles.css` subpath, header logo, and button sizes from Studio for host-app integration.

Moves all Studio panels onto a dockable, persisted layout that remembers arrangement between sessions.

Adds the Renders panel to the shared primitives in Studio.

v0.8.52 NOTES ↗ STABLE

HyperFrames v0.8.52 adds host-app overlay/stage APIs, importable render queue, 3D-motion registry blocks, and richer render-event metadata.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.8.52 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.52
```

Render events now carry the browser's build and install-path shape, giving CLI consumers richer metadata per render.

Host apps can place overlays on the preview and restyle its stage.

Host apps can import the render queue directly from the package.

Adds seven 3D-motion pieces as installable registry blocks to the Catalog.

Scripts now fail a PR whose After capture is a Before capture.

+2 more

v0.8.51 NOTES ↗ STABLE

HyperFrames v0.8.51 adds `hyperframes timeline`, segmented capture with `--resume` / `--keep-segments`, `HF_SEGMENTED_CAPTURE`, and a Chrome memory sampler.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.8.51 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.51
```

└──▷ **TRY IT**

Inspect all tracks and clips in a project before rendering to verify timeline structure from the command line.

```
$ hyperframes timeline
```

Resume an interrupted long-form render without re-capturing already-completed segments, preserving disk-written frames.

```
$ hyperframes render --resume --keep-segments
```

Route long renders through segmented capture in CI without changing render command flags — set once in the environment.

```
$ HF_SEGMENTED_CAPTURE=1 hyperframes render
```

Adds `hyperframes timeline` CLI subcommand to print a project's tracks and clips.

Adds `--resume` and `--keep-segments` flags to the render command for segmented capture, allowing interrupted long renders to be continued.

Adds `HF_SEGMENTED_CAPTURE` environment variable to route long renders to segmented capture.

Adds `streamingEncodeDurationCapEnabled` engine config flag (default off) to gate streaming encode duration capping.

Emits Chrome memory, capture path, and segment fields on render events in the CLI.

+9 more

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.20.0 NOTES ↗

RANK

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

> OpenAI's TypeScript SDK v7.20.0 adds several new API surfaces: vault-based credential sourcing, external storage configuration, safety case retrieval, and SIP call security fields.

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

01 **Vault credentials via environment variables** NEW 30

Adds environment-variable vault credentials to the API, enabling secrets to be sourced from a vault via environment variables.

— No config key or flag name given, just prose description v7.20.0

02 **SIP media security fields on call events** NEW 30

Adds SIP media security fields to incoming call events in the API.

— Names the event type but no field names or schema v7.20.0

03 **External storage configuration management** NEW 20

Adds external storage configuration management to the API for controlling where data is stored.

— Vague, no endpoint or config surface named v7.20.0

04 **Safety case retrieval endpoint** NEW 20

Adds safety case retrieval to the API, allowing clients to fetch safety case records.

— No endpoint path or method given v7.20.0

v7.20.0 NOTES ↗ LATEST STABLE

> OpenAI Node SDK v7.20.0 adds vault credentials, external storage config, safety case retrieval, safety webhooks, and SIP media security.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v7.20.0 https://github.com/openai/openai-node.git
# already have the repo? check out this version:
$ git checkout v7.20.0
```

Adds environment-variable vault credentials to the API, enabling secrets to be sourced from a vault via environment variables.

Adds external storage configuration management to the API for controlling where data is stored.

Adds safety case retrieval to the API, allowing clients to fetch safety case records.

Adds SIP media security fields to incoming call events in the API.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

llmfit

Feedback ⓘ v1.1.16 NOTES ↗ CODE ↗ RANK

Hundreds of models & providers. One command to find what runs on your hardware.

> llmfit's v1.1.16 release adds a disk-planning subcommand for model libraries, native support for 1.58-bit (ternary) models via bitnet.cpp, concurrent-session capacity estimation, and a September 2026 model catalog refresh.

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

01 **Disk planning subcommand for model libraries** NEW 90

A new `storage` subcommand sizes a drive for a local model collection, with flags `--keep` (number of models, default 3), `--selection` (`score` | `largest`), `--os-reserve` (default `100G`), and `--scratch` (default `auto`).

— Named subcommand with all flags and defaults given v1.1.16

THINNER COVERAGE BELOW

02 **Native 1.58-bit (ternary) model support** NEW 50

Adds recognition and correct sizing of native ternary (1.58-bit) models, routing them to the `bitnet.cpp` runtime.

— Explains what and routing target, no usage steps v1.1.16

03 **September 2026 model catalog refresh** IMPROVED 35

Adds a September 2026 model catalog refresh including GLM-5.3, Qwen3.8-Flash-Next, DeepSeek-V4.1, Kimi-K3, and more.

— Names specific models but no mechanism v1.1.16

04 **Concurrent-session capacity estimation** NEW 10

llmfit v1.1.16 adds concurrent-session capacity estimation, per the release summary.

— Mentioned only in summary, no detail on mechanism v1.1.16

v1.1.16 NOTES ↗ CODE ↗ LATEST STABLE

llmfit v1.1.16 adds concurrent-session capacity estimation, disk planning for model libraries, native 1.58-bit model support, and a September 2026 model refresh.

└──▷ GET THIS VERSION

```
$ git clone --branch v1.1.16
https://github.com/AlexsJones/llmfit.git
# already have the repo? check out this version:
$ git checkout v1.1.16
```

Adds disk planning for model libraries via the `storage` subcommand, with flags `--keep` (number of models, default 3), `--selection` (`score` | `largest`), `--os-reserve` (default `100G`), and `--scratch` (default `auto`) to size a drive for a local model collection.

Adds recognition and correct sizing of native ternary (1.58-bit) models, routing them to the `bitnet.cpp` runtime.

Adds September 2026 model catalog refresh including GLM-5.3, Qwen3.8-Flash-Next, DeepSeek-V4.1, Kimi-K3, and more.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Hardware

Minisforum MS-S1 Max Feedback ⓘ

price change · Sep 19, 2026 STORE ↗

A Ryzen AI Max+ 395 mini workstation with 80 Gbps USB4 v2 and 2U rack-mount support, aimed at multi-unit local AI clusters.

MS-S1 Max · 128 GB `ms-s1-max-128`

$3,799 ~~$2,919.90~~ ▲ $879.10 · 30%

list price at store.minisforum.com · Sep 19, 2026

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Volcengine OpenViking

Feedback ⓘ v0.4.21 NOTES ↗

RANK

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

> OpenViking v0.4.21 tightens MCP `search` parameter validation so context-only fields now require `mode='context'` instead of being silently dropped in list mode, alongside new Hermes/MiMo/WorkBuddy agent integrations, VikingDB glob retrieval, Studio conversation and onboarding features, broader Python 3.8 SDK support, and several breaking changes to Docker deployment paths and default model configuration.

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

01 MCP search parameter validation and mode split

BREAKING

95

MCP `search` calls in default `mode='list'` now raise an argument error instead of silently ignoring context-only parameters — `query_expansion`, `max_tokens`, `quotas`, `purpose`, `detail`, `detail_by_category`, `dedup_turns`, `exclude_uris`, `peer_scope`, `other_peer_penalty`, `other_peer_penalties`, `rewrite`, `rewrite_max_bullets`. Callers assembling context with token limits or URI exclusions must switch to explicit `mode='context'`, or drop those fields for ranked results in list mode.

Assemble context from the memory store with token limits and URI exclusions — now requires explicit `mode='context'` instead of relying on previously silently accepted parameters in list mode.

```python
result = await client.search(
    query="incident timeline",
    mode="context",
    max_tokens=4000,
    exclude_uris=["viking://resources/internal.md"],
)
```

— full parameter list, mode behavior, and a runnable migration example v0.4.21

02 Docker container working directory change

BREAKING

70

Docker deployments: the default container working directory changed from `/app` to `/app/.openviking`; custom scripts relying on `/app` as cwd must use absolute paths or set cwd explicitly.

— exact before/after paths with clear migration instruction v0.4.21

03 **Python 3.8 support in openviking-sdk** IMPROVED 65

Expands `openviking-sdk` Python support from `>=3.10` to `>=3.8`, released as `openviking-sdk==0.1.12`.

— exact version constraints and package release given v0.4.21

04 **Codex setup wizard default model change** BREAKING 60

Changes the Codex OAuth setup wizard default model from retired `gpt-5.4` to `gpt-5.6-terra`; existing saved `gpt-5.4` configs are not rewritten automatically and should be updated during maintenance.

— clear before/after model names but no command to update configs v0.4.21

THINNER COVERAGE BELOW

05 **VikingDB write field size limits** BREAKING 55

Remote VikingDB writes now enforce string/text field size limits; oversized `abstract` values may be stored as prefixes, so callers should not depend on exact matching of oversized abstracts.

— names affected field but gives no numeric size limit v0.4.21

06 **New agent integrations: Hermes, MiMo, WorkBuddy** NEW 45

Adds a standalone Hermes OpenViking memory provider for agent memory integrations, and adds Hermes, `MiMo` / `MiMoCode`, and `WorkBuddy` as new log sources for Agent integrations.

— named integrations but no setup or config details given v0.4.21

07 **Studio conversation, onboarding, and pagination updates** NEW 40

Adds VikingBot conversation support and Feishu onboarding integration in Studio, plus server-side user pagination for user lists in Studio.

— three named additions but no UI path or config given v0.4.21

08 **ACL docs relations capability removed** DEPRECATED 40

The stale `relations` capability wording has been removed from ACL docs; `relations` should no longer be treated as part of the file-operation capability list.

— documentation correction only, no code-level impact described v0.4.21

09 **Remote VikingDB glob support** NEW 25

Adds remote VikingDB glob support for retrieval queries.

— one-line addition with no usage detail or example v0.4.21

└──▷ **BREAKING ON UPGRADE**

! MCP `search` callers: in default `mode='list'`, passing any of `query_expansion`, `max_tokens`, `quotas`, `purpose`, `detail`, `detail_by_category`, `dedup_turns`, `exclude_uris`, `peer_scope`, `other_peer_penalty`, `other_peer_penalties`, `rewrite`, or `rewrite_max_bullets` now raises an argument error instead of being silently ignored. Switch to `mode='context'` for context assembly, or remove those fields for ranked results.

! Docker deployments: the default container working directory changed from `/app` to `/app/.openviking`; custom scripts relying on `/app` as cwd must use absolute paths or set cwd explicitly.

! Codex OAuth setup wizard now defaults to `gpt-5.6-terra`; existing saved `gpt-5.4` configs are not rewritten automatically and should be updated during maintenance.

! Remote VikingDB writes now enforce string/text field size limits; oversized `abstract` values may be stored as prefixes — do not depend on exact matching of oversized abstracts.

! The stale `relations` capability wording has been removed from ACL docs; `relations` should no longer be treated as part of the file-operation capability list.

v0.4.21 NOTES ↗ LATEST STABLE

OpenViking v0.4.21 adds Hermes/MiMo/WorkBuddy integrations, VikingDB glob, Feishu onboarding, and Python 3.8 SDK support.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.4.21
https://github.com/volcengine/OpenViking.git
# already have the repo? check out this version:
$ git checkout v0.4.21
```

└──▷ **USE IT**

Assemble context from the memory store with token limits and URI exclusions — now requires explicit `mode='context'` instead of relying on previously silently accepted parameters in list mode.

```python
result = await client.search(
    query="incident timeline",
    mode="context",
    max_tokens=4000,
    exclude_uris=["viking://resources/internal.md"],
)
```

Adds a standalone Hermes OpenViking memory provider for agent memory integrations.

Adds Hermes, MiMo/MiMoCode, and WorkBuddy as new log sources for Agent integrations.

Adds remote VikingDB glob support for retrieval queries.

Adds MCP `search` parameter validation that rejects context-only parameters (`query_expansion`, `max_tokens`, `quotas`, `purpose`, `detail`, `detail_by_category`, `dedup_turns`, `exclude_uris`, `peer_scope`, `other_peer_penalty`, `other_peer_penalties`, `rewrite`, `rewrite_max_bullets`) when called in default `mode='list'`, instead of silently ignoring them.

Adds VikingBot conversation support and Feishu onboarding integration in Studio.

+3 more

BREAKING ON UPGRADE

- MCP `search` callers: in default `mode='list'`, passing any of `query_expansion`, `max_tokens`, `quotas`, `purpose`, `detail`, `detail_by_category`, `dedup_turns`, `exclude_uris`, `peer_scope`, `other_peer_penalty`, `other_peer_penalties`, `rewrite`, or `rewrite_max_bullets` now raises an argument error instead of being silently ignored. Switch to `mode='context'` for context assembly, or remove those fields for ranked results.
- Docker deployments: the default container working directory changed from `/app` to `/app/.openviking`; custom scripts relying on `/app` as cwd must use absolute paths or set cwd explicitly.
- Codex OAuth setup wizard now defaults to `gpt-5.6-terra`; existing saved `gpt-5.4` configs are not rewritten automatically and should be updated during maintenance.
- Remote VikingDB writes now enforce string/text field size limits; oversized `abstract` values may be stored as prefixes — do not depend on exact matching of oversized abstracts.

> ! The stale `relations` capability wording has been removed from ACL docs; `relations` should no longer be treated as part of the file-operation capability list.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20260920 NOTES ↗

RANK

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

> LangSmith's biggest addition this window is a new `GET /v2/threads/stats` endpoint for self-hosted ClickHouse deployments, alongside Application-value resource tagging, a configurable license-expiration warning window, `ls_provider` tagging for custom models in Gateway traces, and a tightened permission requirement for dataset exports.

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

01 Threads stats endpoint for self-hosted deployments NEW

90

`GET /v2/threads/stats` is now available on self-hosted deployments backed by ClickHouse, including when SmithDB queries is disabled. It can be queried with a `project_id` parameter to pull aggregated latency, token, and cost statistics for a tracing project, e.g. to feed a dashboard or cost-alerting pipeline.

Pull aggregated latency, token, and cost statistics for a tracing project to feed a security-operations dashboard or cost-alerting pipeline.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

— Named endpoint with runnable curl example and backend requirements. snapshot-20260920

02 ls_provider tagging for custom models in Gateway NEW

75

Custom OpenAI-compatible models can now be tagged with `ls_provider` (e.g. `{"model": "my-custom-gpt4", "base_url": "https://my-gateway.example.com/v1", "ls_provider": "openai"}`) so Gateway traces carry the correct provider label and map to provider-specific pricing.

Tag an OpenAI-compatible custom model with `ls_provider` so Gateway traces carry the correct provider label and map to provider-specific pricing.

```json
{
  "model": "my-custom-gpt4",
  "base_url": "https://my-gateway.example.com/v1",
  "ls_provider": "openai"
}
```

— Names the field and gives a full config example. snapshot-20260920

THINNER COVERAGE BELOW

03 **Application tag for resource assignment** NEW 45

Workspace editors can now assign resources to an Application tag value from Settings > Resource Tags.

— Only a UI path given, no further mechanism described. snapshot-20260920

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read AND download permissions — users or API tokens that previously exported with only dataset read permission will receive a forbidden response.

snapshot-20260920 NOTES ↗ LATEST STABLE

LangSmith adds `/v2` /threads/stats endpoint, IAM role-based ECR support, ls_provider tracing, and Gateway fallback timeouts

└──▷ **TRY IT**

Pull aggregated latency, token, and cost statistics for a tracing project to feed a security-operations dashboard or cost-alerting pipeline.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

Set the license-expiration warning banner to appear 14 days early on a self-hosted deployment so ops teams have more runway before the instance goes dark.

```env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Tag an OpenAI-compatible custom model with `ls_provider` so Gateway traces carry the correct provider label and map to provider-specific pricing.

```json
{
  "model": "my-custom-gpt4",
  "base_url": "https://my-gateway.example.com/v1",
  "ls_provider": "openai"
}
```

Makes `GET /v2/threads/stats` available on self-hosted deployments backed by ClickHouse, including when SmithDB queries is disabled.

Workspace editors can now assign resources to an Application tag value from Settings > Resource Tags.

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read AND download permissions — users or API tokens that previously exported with only dataset read permission will receive a forbidden response.

WAS THIS USEFUL? 👍 👎

UK AI Security Institute Inspect AI

Feedback ⓘ

0.3.266 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 shipped a single breaking change in this window: the interim-metrics payload for `inspect ctl task score` restructures how scorer identity and score name are reported.

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

01 Renamed fields in task score metrics payload

BREAKING

75

The `inspect ctl task score` interim-metrics payload renames the `scorer` field: it now holds the originating scorer name instead of the score name, and the score key moves to a new `name` field. Clients that read the old `scorer` field as the score name will instead receive the scorer identity, so integrations parsing this payload need to be updated to read `name` for the score key.

— Names exact fields and command but no migration example given 0.3.266

└──▷ **BREAKING ON UPGRADE**

! The `inspect ctl task score` interim-metrics payload renames the `scorer` field: it now holds the originating scorer name, and the score key moves to a new `name` field. Clients that read the old `scorer` field as the score name will receive the scorer identity instead.

0.3.266 CODE ↗ LATEST STABLE

└──▷ **GET THIS VERSION**

```
$ git clone --branch 0.3.266
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.266
```

└──▷ **BREAKING ON UPGRADE**

! The `inspect ctl task score` interim-metrics payload renames the `scorer` field: it now holds the originating scorer name, and the score key moves to a new `name` field. Clients that read the old `scorer` field as the score name will receive the scorer identity instead.

WAS THIS USEFUL?  

Braintrust

Feedback ⓘ snapshot-20260920 NOTES ↗ RANK

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

> Braintrust expanded tracing to new coding agents and models, added an AI-powered trace debugger, moved Loop to a managed runtime for deeper investigations, and introduced log search optimization alongside a breaking change to provider scoping.

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

01 Metric columnstore for log search optimization NEW

60

A new Metric columnstore option under project Settings > Log search optimization speeds up log filtering and trace sorting on the Logs page for high-volume projects.

Enable Metric columnstore on a high-volume project to speed up log filtering and trace sorting in the Logs page.

📍 In the project, go to Settings › Log search optimization and enable Metric columnstore.

– Exact UI navigation path provided for enabling the feature snapshot-20260920

THINNER COVERAGE BELOW

02 Jev (TypeSafe) model support NEW

55

Adds support for Jev (TypeSafe), a model that returns a choice or numeric result, for tracing calls, scoring outputs, and classifying logs. Requires Python v0.41.0+ or TypeScript v3.34.0+ and data plane v2.15.0+.

– Version requirements given but no usage steps snapshot-20260920

03 Trace Debugger and Analyze trace NEW

55

Debugger reports likely failure modes for a single trace, citing spans, tool calls, and model outputs. Alongside it, Analyze trace groups a run into Work sections. Requires data plane v2.12.0+.

– Describes mechanism but no navigation or command given snapshot-20260920

04 Loop runs in managed runtime IMPROVED

55

Loop now runs in a Braintrust-managed runtime rather than the browser, enabling longer investigations and persistent threads, and letting users create and edit prompts, scorers, datasets, facets, custom views, dashboards, and automations across logs, experiments, and datasets. Requires data plane v2.9.0+.

— Lists many capabilities but no concrete entry point snapshot-20260920

└──▷ **BREAKING ON UPGRADE**

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

snapshot-20260920 NOTES ↗ LATEST STABLE

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

└──▷ **HOW TO FIND IT**

Enable Metric columnstore on a high-volume project to speed up log filtering and trace sorting in the Logs page.

📍 In the project, go to Settings › Log search optimization and enable Metric columnstore.

Adds Jev (TypeSafe) support for tracing calls, scoring outputs, and classifying logs using a model that returns a choice or numeric result; requires Python v0.41.0+ or TypeScript v3.34.0+ and data plane v2.15.0+.

Adds Debugger to report likely failure modes for a single trace, citing spans, tool calls, and model outputs, alongside Analyze trace which groups a run into Work sections; requires data plane v2.12.0+.

Expands Loop to run in a Braintrust-managed runtime (rather than the browser), enabling longer investigations, persistent threads, and the ability to create and edit prompts, scorers, datasets, facets, custom views, dashboards, and automations across logs, experiments, and datasets; requires data plane v2.9.0+.

└──▷ **BREAKING ON UPGRADE**

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

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

agent-safe-pipeline

Feedback ⓘ v0.3.1 NOTES ↗ RANK

Agent Safe Pipeline provides a security-focused framework for building AI agent workflows with policy checks and safety controls.

> Agent Safe Pipeline shipped transparent network interception for agent workloads, published the Agent-Safe Intent v1 specification with conformance vectors and framework adapters, added provider-signed effect receipts (VP-3) with Rust, .NET and Kong Gateway integrations, and began publishing signed runtime images to Docker Hub.

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

01 **Infrastructure-scale Compromised Principal Test example** NEW

95

`pnpm --filter @decionis/agent-safe-example-infra-scale demo` runs the infrastructure-scale Compromised Principal Test — seven attacks (960-replica scale, cross-service, cross-cluster, post-authorization mutation, swapped approval, replayed grant, direct cluster call) against a valid-credential agent, resulting in zero unauthorized effects (source in `examples/infra-scale-demo`); it is now published as a canonical infrastructure example.

— Runnable demo command with concrete attack scenarios and scale numbers. v0.3.0

02 **Signed Docker Hub image distribution** NEW

90

Publishes the runtime container image to Docker Hub as `docker.io/decionis/agentsafe:0.2.1` and `docker.io/decionis/agentsafe:0.2.1-init`, with digest-matched manifests and attestations verifiable via `gh attestation verify oci://docker.io/decionis/agentsafe@sha256:...`.

Verify the Docker Hub copy of the agentsafe runtime image is genuine before deploying it in an air-gapped or Docker Hub-preferred environment.

```
$ gh attestation verify
oci://docker.io/decionis/agentsafe@sha256:f96b60ee398a002cdc9c274
a2116c619ef2a65108561bb73d41c9b3d75718135 --repo decionis/agent-
safe-pipeline
```

— Exact image tags and a runnable verification command. v0.3.1

03 **Transparent traffic interception for agent workloads** NEW

88

`agentsafe intercept` redirects a pod's or container's outbound ports 80 and 443 into AgentSafe at the network layer without reconfiguring the workload, reporting every destination reached (documented in `docs/gateway/transparent-interception.md`). The `ghcr.io/decionis/agentsafe:0.2.1-init` init container adds the same interception alongside the workload, letting operators observe agent traffic without modifying it.

Intercept all outbound traffic from a running pod without touching its configuration, to observe every destination the workload reaches before enforcing policy.

```
$ agentsafe intercept
```

— Names command, ports, doc path and init container tag but no deploy walkthrough. v0.3.1

04 **Offline Agent-Safe Intent v1 conformance verification** NEW

88

`agentsafe verify intent <file|dir>` runs the Agent-Safe Intent v1 conformance corpus offline, or prints the canonical bytes and hash of a binding for cross-implementation comparison; `agentsafe verify intent conformance/vectors conformance/agent-safe-intent-v1.json` is the concrete invocation for reproducing published conformance vectors from their bytes.

Verify your Agent-Safe Intent v1 binding implementation offline against the published conformance vectors to confirm canonical hashing matches the reference.

```
$ agentsafe verify intent conformance/vectors conformance/agent-safe-intent-v1.json
```

— Runnable command with exact paths reproducing published vectors. product docs

05 **Agent-Safe Intent v1 specification and conformance vectors**

NEW

80

Publishes Agent-Safe Intent v1 (`agent-safe.intent/1`): a binding specification with canonical form and hash, a JSON Schema generated from the reference implementation, a changelog and change process, plus edge-case canonical-hash conformance vectors under `conformance/vectors` covering Unicode/astral, NFC vs NFD, negative zero, fractional/exponent numbers, nested arrays, UTF-16 key sort order, and the Compromised Principal Test as a vector (one intent, eight single-field mutations, nine distinct hashes).

— Rich spec detail and named edge cases, but no direct command to act on. product docs

06 **Framework adapter coverage for Intent v1** NEW 70

Adds framework adapter coverage in `conformance/frameworks/` for OpenAI `function_call`, Vercel AI SDK `tool-call`, and LangChain `ToolCall`, each with proposal, trusted context, binding, hash, and a coverage row over eight producer capabilities; the framework matrix lives in `spec/intent/v1/frameworks.md`.

— Names three frameworks and files but no runnable step. product docs

07 **Verifying Provider Profile implementations across languages and gateways**

NEW 70

Verifying Provider Profile support extends across languages and gateways: Rust and .NET implementations of the profile, a Kong Gateway plugin that runs the Go verifier and signs the effect receipt in the response phase, and verified provider integration targets for Envoy `ext_authz`, Kong, Spring, Apigee, the Rust `tower` layer, and ASP.NET Core, each with published conformance vectors and independent verifiers.

— Lists many concrete integration targets but no setup steps given. v0.3.0

08 **Provider-signed effect receipts (VP-3)** NEW 65

Provider-signed receipt support (VP-3) lets a verifying provider return a signed receipt of the effect it executed; the executor forwards it at finalization, and Decionis verifies it against the organisation's registered provider key and records it with the commit. Adapter-layer comparison checks what a provider's receipt states against what the pipeline actually observed, enabling downstream verification of provider-stated vs. observed effects.

— Full mechanism described but no command or config surface named. v0.3.0

09 **Shadow-mode enforcement report via agentsafe status** NEW 60

`agentsafe status` prints the gateway's shadow report — what enforcement would have held or refused so far, broken down by action — while the gateway runs in shadow mode, letting operators gauge impact before switching to enforcement.

Check how many actions enforcement would have blocked in shadow mode before flipping the gateway to enforcement, broken down by action.

```
$ agentsafe status
```

— Command is clear but shadow report's fields aren't detailed. product docs

v0.3.1 NOTES ↗ LATEST STABLE

agent-safe-pipeline v0.3.1 adds transparent network interception beside workloads and Docker Hub image distribution.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.3.1 https://github.com/decionis/agent-safe-pipeline.git
# already have the repo? check out this version:
$ git checkout v0.3.1
```

└──▷ **TRY IT**

Verify the Docker Hub copy of the agentsafe runtime image is genuine before deploying it in an air-gapped or Docker Hub-preferred environment.

```
$ gh attestation verify oci://docker.io/decionis/agentsafe@sha256:f96b60ee398a002cdc9c274a2116c619ef2a65108561bb73d41c9b3d75718135 --repo decionis/agent-safe-pipeline
```

Publishes the runtime container image to Docker Hub (`docker.io/decionis/agentsafe:0.2.1` and `docker.io/decionis/agentsafe:0.2.1-init`) with digest-matched manifests and attestations verifiable via `gh attestation verify oci://docker.io/decionis/agentsafe@sha256:...`.

Adds transparent interception alongside the workload via the `ghcr.io/decionis/agentsafe:0.2.1-init` init container, enabling observation of agent traffic without modifying the workload itself.

v0.3.0 NOTES ↗ STABLE

agent-safe-pipeline v0.3.0 adds Rust and .NET verifier implementations, a Kong Gateway plugin, effect-receipt signing, and a Compromised Principal Test example.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v0.3.0 https://github.com/decionis/agent-safe-pipeline.git
# already have the repo? check out this version:
$ git checkout v0.3.0
```

Adds a Rust implementation of the Verifying Provider Profile.

> Adds a .NET implementation of the Verifying Provider Profile.
>
> Adds a Kong Gateway plugin that runs the Go verifier and signs the effect receipt in the response phase.
>
> Introduces the effect receipt (VP-3): the provider signs what it actually executed, enabling downstream verification of provider-stated vs. observed effects.
>
> Adds adapter-layer comparison of what a provider's receipt states with what the pipeline observed.
>
> +1 more

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

Alibaba Qwen Models

Feedback ⓘ qwen-audio-3.1-realtime-plus

NOTES ↗ RANK

Qwen is Alibaba's language and multimodal model family, including coding, vision, image, speech, and reasoning models.

> Qwen shipped a major update to its real-time duplex speech model, qwen-audio-3.1-realtime-plus, adding new voices, cloning, search, and a much larger context window.

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

01 **Real-time duplex speech upgrades in qwen-audio-3.1-realtime-plus**

NEW 70

The `qwen-audio-3.1-realtime-plus` model adds eight new system voices to the existing roster, supports voice cloning, adds web search capability, provides a 262,144-token context window, and supports function calling — all within real-time duplex speech conversations.

— Names concrete additions and a token limit but no usage steps.

qwen-audio-3.1-realtime-plus

> qwen-audio-3.1-realtime-plus NOTES ↗ LATEST STABLE
>
> qwen-audio-3.1-realtime-plus adds eight new system voices, voice cloning, web search, and a 262,144-token context window for real-time duplex speech.
>
> Adds eight new system voices to the existing voice roster for real-time duplex speech conversations.
>
> Supports voice cloning in real-time duplex speech sessions.
>
> Adds web search capability during real-time duplex speech conversations.
>
> Provides a 262,144-token context window for real-time duplex speech sessions.
>
> Supports function calling in real-time duplex speech conversations.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 20, 2026

Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue