THE AI TOOLCHAIN

NO. 046

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 4, 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 159 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 23 tools

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

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

Three unrelated things stand out today: LiteLLM makes gateway-level LLM security enforceable across requests; OntoCast turns ontology

vim ? help

extraction into a validated, auditable pipeline; magpie adds embeddings/rerank endpoints and tighter routing controls so more agent traffic can share one governed gateway.

GOVERN

Enforce LLM security policy before requests reach tools, models, or agents

LiteLLM added guardrails across request lifecycle stages, fine-grained MCP and A2A access controls, redaction, encryption, login lockdown, rate limiting, and breached-password detection. That moves security checks into the gateway path, so teams do not have to rely on every agent or app implementing its own filtering and access rules correctly.

LiteLLM

BUILD

Validate ontology extraction while the pipeline is still running

OntoCast now exposes vector-based ontology retrieval tuning, SHACL facts validation, an autofix/critic loop, entity-aggregation guards, web-search grounding, HTTP processing endpoints, and a config-auditing CLI. Teams building structured knowledge pipelines can catch invalid facts and bad extraction settings before they become downstream graph cleanup work.

OntoCast

DEPLOY

Route more agent, embedding, and rerank traffic through one gateway

magpie added first-class integrations for VS Code Copilot Chat, JetBrains Air, Mister Morph, and AtomCode, plus gateway `/v1/embeddings` and `/v1/rerank` endpoints and routing controls such as purpose filters, per-group first-token timeouts, and per-account usage caps. That lets teams centralize more AI traffic instead of managing separate controls per IDE agent or model endpoint.

magpie

OPERATE

Keep agent and serverless failures diagnosable after retries and scaling events

Zeroshot now preserves node failure explanations across execution environments, so retries and parallel branches do not overwrite the original cause. Modal added Endpoint visibility, affinity-based request

> routing, and finer container metrics, giving teams more of the context needed to understand where live inference or agent workloads are slowing down or failing.
>
> Zeroshot · Modal

Does **GPT-5.5** read better? 👍 👎

BUILD

◆ AI Coding Agents

yetone magpie

Feedback ⓘ v0.1.859 CODE ↗ RANK

magpie is a macOS menu-bar app, TUI, and CLI that lists every AI coding agent on the machine and switches each one's model, such as Codex on DeepSeek or Claude Code on Kimi, by rewriting only that key in the agent's own config.

> Magpie's busiest window added four new first-class agent integrations — VS Code Copilot Chat, JetBrains Air, Mister Morph, and AtomCode — plus new gateway `/v1/embeddings` and `/v1/rerank` endpoints, a run of CLI additions (`magpie quota wait`, `magpie plugin add`, proxy/mirror flags, a Docker NAS gateway mode), and finer routing controls spanning a purpose filter on usage/routing, per-group first-token timeouts, and per-account usage caps.

└──▷ **WHAT SHIPPED · 36 FEATURES** most completely described first ? what's the number?

01 Purpose filter on Usage and Routing pages NEW 95

Adds a purpose picker to the Usage page filtering token/cost totals, charts, and both CSV export paths before pagination, with choices drawn from the whole period. Adds a `purpose` query parameter to the `/api/usage/requests` endpoint accepting values like `kind:thread_title`, `kind:guardian`, `kind:memory_consolidation`, `kind:collab_spawn`, `kind:ambient_suggestions`, or `unmarked`. The Routing page gains the same picker for its live/historical request list, session summaries, and replay selection, clearing itself when navigating to a request outside the active filter.

Pull only title-generation requests for today from the Usage API to audit background LLM spend without conversation traffic.

```
$ curl 'http://localhost:<port>/api/usage/requests?
period=today&purpose=kind:thread_title&agent=codex&limit=50&offse
t=0'
```

Isolate records that carry no kind at all — logged as `Unmarked` — to spot unclassified requests across the current period.

```
$ curl 'http://localhost:<port>/api/usage/requests?
period=today&purpose=unmarked&limit=50&offset=0'
```

— Endpoint, param values, and runnable curl examples all given v0.1.854

02 Mister Morph agent integration NEW 90

Adds `~/.morph/config.yaml` (or `$MISTER_MORPH_CONFIG`) integration: selecting a magpie model sets `llm.inference_provider`, `llm.endpoint`, `llm.api_key`, `llm.model`, `llm.context_window_tokens`, and `llm.headers.User-Agent` for Mister Morph (`morph`), configured as an `openai_response_compatible` gateway using the Responses API, and restores previous values when the user switches away.

Point Mister Morph at the magpie gateway for a specific model by letting magpie write its config — use a non-default config path when needed.

```
$ export MISTER_MORPH_CONFIG=~/.config/morph/config.yaml
```

Inspect the keys magpie writes into the Mister Morph config to verify gateway routing is correct.

```yaml
llm:
  inference_provider: openai_response_compatible
  endpoint: https://<gateway>/v1
  api_key: <gateway-token>
  model: deepseek/pro
  context_window_tokens: 65536
  headers:
    User-Agent: mistermorph
```

— Config keys and file path shown with a working example `v0.1.801`

03 **AtomCode first-class agent support** NEW 90

Writes one `[provider_accounts."magpie"]` table (with `provider = "openai-compatible"`, verified against AtomCode 5.2.1, matching the shape AtomCode's own CodingPlan sign-in writes since v5.0.7) and one `[models."magpie/<provider>/<model>"]` table per catalog model into `~/.atomcode/config.toml` (or `$ATOMCODE_HOME`); sets `default_provider` and `default_model` to the picked model, stashes the previous defaults and restores them on exit — no gateway changes required, as AtomCode speaks plain Chat Completions with `User-Agent: atomcode/<version>` and Bearer auth.

The shape magpie writes to AtomCode's config — one provider account pointing at the magpie gateway and one model entry per catalog model.

```toml
default_provider = "magpie/<provider>/<model>"
default_model = "magpie/<provider>/<model>"

[provider_accounts."magpie"]
provider = "openai-compatible"
base_url = "http://127.0.0.1:<port>/v1"
api_key = "…-atomcode"

[models."magpie/<provider>/<model>"]
account = "magpie"
model = "<provider>/<model>"
```

Wire AtomCode to magpie's full model catalog so all AtomCode requests are attributed and priced correctly in Usage.

```toml
[provider_accounts."magpie"]
provider = "openai"
base_url = "http://127.0.0.1:<port>/v1"
api_key = "<magpie-atomcode-token>"

[models."magpie/<provider>/<model>"]
account = "magpie"
model = "<provider>/<model>"
context_window = 128000

default_provider = "magpie/<provider>/<model>"
default_model = "magpie/<provider>/<model>"
```

— Exact config.toml shape shown with a runnable example v0.1.818 v0.1.820

04 **magpie quota and quota wait CLI commands** NEW 90

Adds `magpie quota` to display remaining allowance for every subscription, plan, and key balance, and `magpie quota wait <subscription>` which blocks until the named subscription or account has allowance again, then exits 0, enabling unattended retry loops. Adds `--timeout <duration>` and `--quiet` flags to `magpie quota wait`; it exits 1 on timeout, 2 on unknown name, 130 on Ctrl+C.

Block a script until a Codex account has quota again, then resume automatically — useful for long unattended tasks that hit rate limits.

```
$ until codex exec "refactor all tests"; do magpie quota wait
codex || break; done
```

Block a long agentic task until a Codex quota resets, then resume it unattended — useful in CI or overnight runs.

```
$ until codex exec "…"; do magpie quota wait codex || break; done
```

— Exact command, flags, exit codes, and runnable loop example `product docs`

05 **Docker NAS gateway deployment** NEW 90

Adds Docker container deployment for headless NAS gateway use via `ghcr.io/yetone/magpie:latest`, configured via `MAGPIE_ADDR`, `MAGPIE_PUBLIC_URL`, and `MAGPIE_WEB_KEY` environment variables and a bind-mounted `/config` directory (must be writable by uid 65532), with the web UI served on port 3430 and the gateway on port 3425.

Deploy magpie as a shared NAS gateway so agents on other machines can route through it.

yaml

```yaml
services:
  magpie:
    image: ghcr.io/yetone/magpie:latest
    restart: unless-stopped
    ports:
      - "127.0.0.1:3425:3425"
      - "127.0.0.1:3430:3430"
    environment:
      MAGPIE_ADDR: 0.0.0.0:3425
      MAGPIE_PUBLIC_URL: http://<nas-ip>:3425
      MAGPIE_WEB_KEY: <your-key>
    volumes:
      - ./config:/config
    command: [web, --addr, 0.0.0.0:3430, --no-open]
```

Run magpie as a shared NAS gateway so agents on other machines can reach it over the LAN.

```yaml
services:
  magpie:
    image: ghcr.io/yetone/magpie:latest
    restart: unless-stopped
    ports:
      - '127.0.0.1:3425:3425'
      - '127.0.0.1:3430:3430'
    environment:
      MAGPIE_ADDR: 0.0.0.0:3425
      MAGPIE_PUBLIC_URL: http://<nas-ip>:3425
      MAGPIE_WEB_KEY: <key>
    volumes:
      - ./config:/config
    command: [web, --addr, 0.0.0.0:3430, --no-open]
```

— Env vars, ports, and a runnable docker-compose example `product docs`

06 **Gateway embeddings and rerank endpoints** NEW 90

Adds `POST /v1/embeddings` and `POST /embeddings` endpoints, proxying OpenAI-shaped embedding requests to the model's own provider without a separate base URL or key. Adds `POST /v1/rerank` and `POST /rerank` endpoints, proxying rerank requests in the Cohere/Jina/Voyage shared shape to the model's own provider, with usage recorded and tokens counted like a chat turn.

Send an embedding request through magpie so a local RAG stack uses one base URL and one key for both chat and embeddings.

```
$ curl -X POST http://localhost:8080/v1/embeddings \
  -H 'Authorization: Bearer <your-magpie-key>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "<magpie-model-id>", "input": ["hello world",
"semantic search"]}'
```

Rerank candidate passages for a retrieval pipeline without a second provider credential — magpie forwards to the model's own provider.

```
$ curl -X POST http://localhost:8080/v1/rerank \
  -H 'Authorization: Bearer <your-magpie-key>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "<magpie-model-id>", "query": "what is magpie?",
"documents": ["magpie is a gateway", "a bird"]}'
```

— Endpoints and runnable curl requests both given v0.1.825

07 **Alma model capability overrides** NEW 90

Adds `capabilityOverrides` to the `availableModels` entries in the `PUT /api/providers/:id/models` payload, letting Alma substitute a model's known capabilities (including reasoning levels such as `xhigh`, `minimal`, `low`, `medium`, `high`, `ultra`) with values magpie supplies — so thinking is now offered for every Codex model.

Override capability metadata for a Codex model so Alma surfaces all reasoning effort levels (e.g. after adding a custom provider whose models are unknown to models.dev).

```
$ curl -s -X PUT http://localhost:
<port>/api/providers/<id>/models \
  -H 'Content-Type: application/json' \
  -d '{"models":["gpt-5.5"],"availableModels":[{"id":"gpt-
5.5","name":"GPT-5.5","capabilityOverrides":{"efforts":
["xhigh","minimal","low","medium","high","ultra"]}}]}'
```

— Endpoint, payload field, and runnable curl example given v0.1.802

08 **VS Code Copilot Chat integration** NEW 85

Registers a `{"vendor": "customendpoint", "name": "magpie"}` group in `chatLanguageModels.json` (beside VS Code's `settings.json` in its User folder, e.g. `~/Library/Application Support/Code/User/`), listing catalog models at the gateway's `/v1/chat/completions` with tool calling enabled — no GitHub sign-in, Copilot plan, or API key entry required, as magpie's token is injected via each model's `requestHeaders` Authorization field; requires VS Code 1.122+. Also sets `chat.defaultModel` in `settings.json` to the selected catalog model ID on connect, restoring the previous value on disconnect.

— Exact files and settings keys named, no runnable command v0.1.803

09 **Installer and update proxy/mirror support** NEW 85

Adds `--proxy <url>` flag (also `MAGPIE_PROXY`, `HTTPS_PROXY`, `ALL_PROXY`) and `--mirror <url>` flag (also `MAGPIE_MIRROR`) to the installer and `magpie update` for firewall or GitHub-restricted environments, with `magpie update mirror <url>|off` to persist a GitHub download mirror for every subsequent update including the app, while downloads still verify SHA-256 from usemagpie.ai.

Update magpie behind a corporate firewall or in a GitHub-restricted environment by routing through an HTTP proxy.

```
$ magpie update --proxy http://127.0.0.1:7890
```

— Flags, env vars, subcommand, and a runnable example given `product docs`

10 **First-token timeout and latency breakdown for routing groups** NEW 80

Adds `Group.FirstToken` config field: a group abandons a member that hasn't produced its first content token within the set number of seconds and immediately tries the next member. Also splits first-token latency in request details into magpie's own time versus the vendor's time, giving per-segment TTFT visibility.

Cap how long a slow provider can delay a group response before the next member is tried — useful when one vendor occasionally stalls at stream start.

```yaml
FirstToken: 5
```

— Config key and example value given for direct use `v0.1.844`

11 **JetBrains Air integration** NEW 80

Writes a Magpie ACP agent entry into `acp.json` and `magpie-opencode.json` in Air's config folder (`~/Library/Application Support/JetBrains/Air` on macOS, `$XDG_CONFIG_HOME/JetBrains/Air` on Linux, `%APPDATA%\JetBrains\Air` on Windows), backed by OpenCode's ACP server (`opencode acp`), so Air's agent menu lists Magpie and its model menu shows magpie's models and routing groups.

— Per-OS config paths and backing mechanism named precisely `v0.1.805`

12 **Claude Code internal call optimizations** IMPROVED 70

Passes `--thinking disabled` to the Claude CLI when `ThinkOff` is set on a request, preventing Claude Code's auto-mode classifier and small internal asks from spending thousands of output tokens on thinking. Also routes Claude Code's Haiku-tier internal calls (session titles, Explore subagents, WebFetch reads) to the Haiku model served by the same provider as the main model, rather than the full Opus/Sonnet model.

— Mechanism and flag named, but applied automatically not by user `v0.1.853`

13 **272K context cap for Codex and Claude Code** NEW 70

Writes `CLAUDE_CODE_AUTO_COMPACT_WINDOW` (`claudeCompactEnv`) to cap auto-compaction at the 272K working window for large-context models, removing it when the `fullContext` setting is enabled or the user has their own value.

— Env var and opt-out setting named, no exact UI path `v0.1.836`

14 **magpie plugin add for OpenCode and pi packages** NEW

70

Adds `magpie plugin add <pi-package>` support for pi packages (identified by a `pi` manifest, `pi-package` keyword, or `@earendil-works/pi-coding-agent` dependency, using `pi.registerProvider`), loading them via Bun alongside existing OpenCode provider plugin support; pi's own files are stored under magpie's config directory in `pi/` unless `PI_CODING_AGENT_DIR` overrides.

— Command and override env var named, no worked example `product docs`

15 **Library MCP server address-based discovery** NEW 70

Adds address-based discovery to the Library MCP server search: paste a repository URL (GitHub, GitLab, Bitbucket, Codeberg, Gitee), a live endpoint, an npm package (`@playwright/mcp`), a PyPI project, or a container image (`ghcr.io/…`, `docker.io/…`) to find the matching server directly, without knowing its display name.

— Clear input types and an example package name given `v0.1.800`

16 **Library instruction sync for Droid and Grok Build** NEW

65

Syncs shared Library instructions into Droid's user-wide `~/.factory/AGENTS.md` and Grok Build's user-wide `$GROK_HOME/AGENTS.md` (falling back to `~/.grok/AGENTS.md`), appending them after the user's own content in both.

— Exact file paths given, no command or trigger shown `v0.1.855`

17 **Japanese localization across UI, tray, and docs** NEW

65

Adds Japanese (`ja`) as a selectable language in Settings › Language with system-language auto-detection following any `ja-prefixed` locale; localises the tray menu via `trayJa` (labels like `magpie を開く`, `バージョン %s`, restart/quit strings); adds Japanese quota-alert formatting via `alertWindowJa` and `alertClockJa` (reset clocks as '明日', weekday names 月曜–日曜, Japanese-style dates); and adds Japanese-locale screenshots (`-ja` variants) to the docs site for Getting Started, Intent Routing, and Import pages.

— Settings path and variable names given, no screenshots attached `v0.1.855` `v0.1.851`

18 **Gemini Google sign-in search and image generation** NEW

65

Adds Google Search routing for Gemini models on a Google sign-in (Gemini CLI or Antigravity account): tool-free requests use the provider's native `googleSearch` via Code Assist, while requests with function tools fall back to magpie's own search. Adds image-generation support for Google sign-in accounts via Code Assist, routing draw

requests through `drawCodeAssist` for image models (e.g. `gemini-3.1-flash-image`) listed under the account.

– Named internal routes but behaviour is entirely automatic v0.1.821

19 **Per-account usage caps for routing** NEW 65

Adds `SetAccountCap` to let routing treat a subscription account as used up once it reaches a configured percentage (1–99, via `MinCap` / `MaxCap` constants) of its usage window, moving on to the next account until the window renews. Adds `CapHeld` on `Allowance` to check whether any of an account's usage windows have reached or passed the cap, returning the current usage and renewal time.

– Internal API named precisely, no user-facing config shown v0.1.809

20 **Cloudflare Workers AI decision models: Clef and Clef Flash** NEW 63

Adds `@cf/cloudflare/clef` and `@cf/cloudflare/clef-flash` as decision models on Cloudflare Workers AI, routed via their own `…/ai/run/@cf/cloudflare/clef` and `…/ai/run/@cf/cloudflare/clef-flash` endpoints alongside the existing `typesafe/jev` model.

– Named endpoints and models but no usage example shown v0.1.859

21 **Tencent Cloud preset with per-region overrides** NEW 60

Adds Tencent Cloud as a single preset with three regions (Plan · China, Pay as you go · China, Pay as you go · Global). Adds `Website`, `Catalog`, and `Models` fields to the `Region` struct in presets for per-region overrides, and extends `presetAt` to return the matched region so imported providers get region-specific endpoints, pages, and catalog entries instead of only top-level preset values.

– Internal struct fields named, no direct user action described v0.1.845

22 **Provider editor: manual model entry and icon search** NEW 60

Adds an `Add model` button to manually add models by ID (comma-separated for multiple) when a provider's list doesn't include them. Expands built-in provider icons with 92+ additional vendor/platform marks (including `ai21`, `aws`, `github`, `google-color`, `huggingface-color`, `perplexity-color`, and more from lobehub/icons), selectable via a new `Find an icon…` search field.

– UI buttons named, specific icon names listed, no API v0.1.841

23 **WSL fallback for Claude Code on Windows** NEW 60

When magpie runs on Windows and finds no local `claude` binary, it automatically discovers and runs the `claude` installed in a running WSL distro (via `wslrun`), using

that distro's own home, login, and PATH.

– Mechanism named but entirely automatic, nothing to invoke `v0.1.834`

---

24 **Remote magpie support** NEW 60

Adds Remote magpie support via the app's Add sheet or TUI Providers page (`a` to add, `w` to change address, `m` to fetch its model list), enabling a shared magpie instance to serve agents running on other machines.

– TUI keys named clearly, no further mechanism detail `product docs`

---

25 **Codex internal call labeling in Routing and Usage** IMPROVED 60

Labels unnamed Codex background worker groups in Routing → By session as `Background memory task` / `后台记忆整理` instead of a bare UUID, with a tooltip explaining the worker is consolidating memories from earlier chats. Codex `title_reconsideration` and `title_generation` calls are now tagged `Title` in Routing and Usage instead of their raw internal names; Codex can read a title from a magpie model when `model_provider = "magpie"` is set, and a title answer containing no title is shown as failed.

– Config key named, labeling logic explained, no example run `v0.1.818` `v0.1.814`

THINNER COVERAGE BELOW

---

26 **Hide menu bar icon while keeping usage display** NEW 55

Adds a `trayNoBird` preference, toggled via Settings → Usage → `Menu bar magpie icon` row, to hide the magpie bird icon from the macOS menu bar while still displaying provider usage allowances.

– Exact settings path given; thin beyond that `v0.1.848`

---

27 **WorkBuddy daily check-in command** NEW 55

Adds `magpie accounts checkin` CLI command (also `c` in the TUI's Usage page) to trigger WorkBuddy daily check-in for all signed-in accounts at once.

– Exact command and TUI key named, nothing further `product docs`

---

28 **Copilot model fallback to chat completions** IMPROVED 55

Adds automatic fallback to `/v1/chat/completions` for Copilot models (GPT-4o, GPT-4.1, Education/Student plan models) that list no `supported_endpoints` and are rejected by `/v1/messages`, replacing a misleading plan-refusal error with a successful response.

– Endpoints and models named, fix applied automatically `v0.1.817`

29 **RTK upgrade diagnostics** IMPROVED 55

Surfaces `gainErr` and `codexSandbox` fields in RTKView JSON so callers can distinguish a failed `rtk gain` read from a zero-savings result and see which Windows sandbox Codex runs commands in. RTK upgrade failures now report the tool name, exit code, and last output on the RTK card, and a finished upgrade is confirmed on-card.

— Fields named but informational only, no user action v0.1.814

30 **Agent connection reason reporting** IMPROVED 55

Adds ConnectHow() returning a Connection struct with a How field reporting how the starting model was chosen when an agent connects — values include `'kept'`, `'again'`, `'joined'`, `'magpie'`, `'same'`, `'alike'`, `'default'`, and `'first'`. The Agents UI shows a human-readable explanation using connectedSaid() and `CONNECTED_HOW` templates; the model picker also indicates routing direction and prompts confirmation before switching an agent off magpie.

— Named values and functions but purely informational feature v0.1.806

31 **Codex usage counting accuracy** IMPROVED 55

Counts Codex `token_usage_record` and compaction responses (including those inside `compacted` blocks and paginated files without a history parent) so previously omitted billable calls now appear in session totals. Adds `ResponseID` field (`response_id`) to the Call struct in session records, enabling per-response deduplication of Codex usage.

— Named field and mechanism but no user-facing action v0.1.801

32 **Accurate context window display and copy model ID** IMPROVED 50

Provider model list now reports the effective context window agents are told — the user-set value if configured, otherwise the models.dev window for prefixed model IDs — so the Providers page and agent model pickers agree. Right-clicking a model chip in a provider's editor now includes a copy model ID menu option.

— Clear UI interaction named, no config surface involved v0.1.816

33 **Provider rail in agent model picker** IMPROVED 45

Adds a provider rail to the agent model picker: when three or more provider groups are present and the viewport is at least 600 px wide, providers appear in a column on the left and selecting one shows only that provider's models, eliminating long scroll.

— Thresholds given but pure UI, no config surface v0.1.824

34 **Routing page interaction improvements** IMPROVED 40

Clicking an open routing group's heading now folds it back into its row (matching how it was opened), with a guard that blocks folding when unsaved changes are present. Failed

requests whose route has no order can now be clicked to expand on the Routing page's live list.

— UI-only behaviour, no config or API surface named `v0.1.858` `v0.1.814`

35 **OpenCode reasoning capability propagation** IMPROVED

40

Propagates the Reasoning field into the `opencode.json` provider entry, setting `reasoning: true` for models that think with no effort levels or that support exactly `low`, `medium`, and `high` effort levels, so OpenCode 1's model tooltip correctly shows 支持推理 instead of 不支持推理 for models like mimo-v2.6-flash-free.

— Named field and example model, but automatic and internal `v0.1.797`

36 **Agents page CLI refresh and install listing** IMPROVED

35

Updates the Agents page to refresh every agent's CLI simultaneously and list agents not installed on the current machine alongside their vendors' install commands.

— Describes the UI change only, no concrete surface named `v0.1.801`

NEW INTEGRATION OpenRouter as a System One decision provider

v0.1.838 CODE ↗ STABLE

Adds AsksDecideModels() so System One / Vercel gateway decision providers can test each model individually; `ModelsURL` is now fetched verbatim for decision-only providers.

└──▷ GET THIS VERSION

```
$ git clone --branch v0.1.838 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.838
```

Adds AsksDecideModels() provider method so decision providers using System One or Vercel gateways (including OpenRouter) can have each of their models sent an individual System One test question, rather than being blocked by the blanket `decide` guard.

Adds OpenRouter as a System One decision provider, fetching its decision-model list from the user-supplied `ModelsURL` (e.g. `.../models?output_modalities=decisions`) verbatim — query string and all — instead of the default `/models` endpoint.

WAS THIS USEFUL?  

Graphify Labs Graphify

Feedback ⓘ v0.9.75 NOTES ↗

RANK

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

> Graphify v0.9.75 significantly deepens its code-graph extraction, adding intra-class call binding across five languages, new structural extractors for Zig, PHP, Elixir and Rust, Scala type-reference tracking, and improved bash, Dart and diagnostic handling.

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

01 **Intra-class call binding across languages** NEW 70

Graphify now binds `self` / `this` / `super` member calls to the caller's own class chain across Python (`self.` / `cls.` /super()), JavaScript/TypeScript (this.m()/super.m()), Swift (`self` / `super` , class- and struct/extension-aware), and Ruby (`self` `-sends` within the class chain). Multiple-inheritance ties and unknown ancestors fail closed, producing no fabricated edge.

— Mechanism and scope per language fully described, but no user-facing command. v0.9.75

02 **New language-structure extractors for Zig, PHP, Elixir, Rust**

NEW 70

Adds structural extraction for Zig error-set declarations and their members (emitting `case_of` edges), PHP enum cases both pure and backed (also emitting `case_of` edges), Elixir `defmacro` / `defmacrop` / `defguard` / `defguardp` definitions, and Rust `macro_rules!` definitions with local invocation resolution (standard library macros like `println!` fail closed).

— Names every language, keyword and edge type; no direct invocation step. v0.9.75

03 **Scala type-reference coverage** NEW 70

Scala file-scope `val` / `var` type annotations now emit `references` edges, `type` alias/definition right-hand sides (including `opaque type` and match types) are walked, and context bounds (e.g. `def f[A: Ordering]`) resolve to `references` edges.

— Concrete syntax and edge types named, but automatic not user-invoked. v0.9.75

04 **Single-implementer interface dispatch for Swift protocols**

IMPROVED 65

Single-implementer interface dispatch, previously limited to other languages, is lifted into a language-agnostic core and applied to Swift protocol requirements: calls dispatch

to the sole conformer (tagged `INFERRED`) and fail closed when two or more types conform.

— Clear before/after and tagging behavior, no user action required. v0.9.75

05 **Bash cross-script invocation edges** NEW 65

Bash scripts invoking another script through a non-shell interpreter (e.g. `python3 build.py`, `node build.js`, `"$PYTHON" stage.py`) are now recorded as an `invokes` edge; `$VAR` or path-based command words are not mistaken for an interpreter.

— Concrete examples and edge name given, but passive detection only. v0.9.75

06 **Dart part-file and conditional import linking** NEW 60

Dart `part` files are now linked to their library via a file node, and every URI branch of a configurable import/export (e.g. `import 'a.dart' if (...) 'b.dart';`) is connected rather than only the default branch.

— Names the Dart constructs covered, no further mechanism detail. v0.9.75

07 **Warning for symbol-less parsed files** NEW 60

Graphify now warns once at extraction time when a cleanly-parsed code file yields no symbols beyond its own file node (likely a data literal), while staying silent on empty/whitespace-only files, languages with no extractor, and intentionally-skipped data.

— Describes exact trigger and exceptions, giving a clear diagnostic signal. v0.9.75

THINNER COVERAGE BELOW

08 **Elixir unqualified-call resolution** NEW 55

Elixir unqualified calls now resolve to functions brought into scope via `import` / `use`, scoped per module, failing closed when no in-scope module defines the name.

— States scoping and fail-closed rule but no further detail. v0.9.75

WAS THIS USEFUL? 👍 👎

Earendil Works Pi

Feedback ⓘ v1.0.2 NOTES ↗ RANK

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

> Pi's latest releases focus on model control and extensibility: per-thinking-level sampling overrides, native image generation, Cloudflare Clef classifier models, and a wave of MCP authentication and per-project configuration options, alongside several breaking changes to TUI defaults, OAuth credential storage, and npm packaging.

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

01 Per-thinking-level sampling parameter overrides NEW

95

The `samplingParamsByThinkingLevel` config key (settable in `models.json` or `.pi/config.yaml`) lets OpenAI-compatible APIs (`openai-completions`, `openai-responses`, `azure-openai-responses`) override sampling params like `temperature` and `top_p` per Pi thinking level (`off`, `minimal`, `low`, `medium`, `high`, `xhigh`, `max`), merged on top of the base `samplingParams` with later values winning per key.

Override sampling temperature only at the `high` thinking level, leaving other levels at model defaults — useful when you want more deterministic outputs at lower reasoning intensities.

```json
{
  "samplingParams": {
    "temperature": 0.7,
    "top_p": 0.9
  },
  "samplingParamsByThinkingLevel": {
    "off": {
      "temperature": 0.3,
      "top_p": 0.8
    },
    "high": {
      "top_k": 50
    }
  }
}
```

Tune sampling per thinking level so lighter reasoning modes use higher temperature while deep-think modes stay deterministic.

```json
{
  "samplingParamsByThinkingLevel": {
    "low":    { "temperature": 1.0, "top_p": 0.95 },
    "medium": { "temperature": 0.7, "top_p": 0.9  },
    "high":   { "temperature": 0.2, "top_p": 0.8  }
  }
}
```

Cap token cost for `low` thinking requests while allowing higher temperature for `high` thinking on an OpenAI-compatible endpoint.

```yaml
samplingParamsByThinkingLevel:
  low:
    temperature: 0.3
    top_p: 0.9
  high:
    temperature: 0.7
    top_p: 1.0
```

— Config key, file location, and three runnable examples given `v1.0.2`

02 **Image generation via models.generateImages** NEW 90

Adds image generation support via models.generateImages(model, { input }), surfacing OpenRouter image models such as `google/gemini-2.5-flash-image` and `black-forest-labs/flux.2-pro`; results return base64 image blocks attached via image() and are not saved to disk.

Generate an image from a text prompt using an OpenRouter image model and attach the result so the model can see it.

```javascript
const result = await models.generateImages('google/gemini-2.5-flash-image', {
  input: [{ type: 'text', text: 'A red fox in the snow, watercolor' }]
});
if (result.stopReason !== 'stop') throw result.errorMessage;
for (const block of result.output) {
  if (block.type === 'image') image(block);
}
```

— Runnable code example demonstrates full usage `product docs`

03 **Per-project MCP server overrides via .pi/mcp.json** NEW

88

`.pi/mcp.json` project-level overrides let you enable, disable, or change `enabled`, `exposure`, and `toolExposure` of a user-level MCP server per project; `/mcp` also gains enable/disable toggles for the current project.

Disable a user-level MCP server for one project without touching the global config.

```json
{
  "my-user-server": {
    "enabled": false
  }
}
```

Disable a noisy user-level MCP server for one project without touching the global config.

```json
{
  "my-global-server": {
    "enabled": false
  }
}
```

— Config file path with two worked examples v1.0.1

04 **Expanded MCP server authentication options** NEW

80

Adds top-level Radius sign-in via `"auth": { "provider": "radius" }`, which offers to configure the Radius MCP server after sign-in; `"auth": { "provider": "<provider>" }` for HTTP MCP servers to send a provider's current token as the bearer token instead of using MCP OAuth; Anthropic workload identity federation via `ANTHROPIC_FEDERATION_RULE_ID`, `ANTHROPIC_ORGANIZATION_ID`, and `ANTHROPIC_IDENTITY_TOKEN_FILE` environment variables; `oauth.authServerMetadataUrl` for MCP servers that advertise a wrong or missing OAuth authorization server; and `oauth.clientRegistration: "cimd"` so authorization servers can recognize pi by its Client ID Metadata Document URL instead of dynamic registration.

Allow an OAuth-gated MCP server to recognise pi by its published Client ID Metadata Document instead of requiring dynamic client registration.

```json
{
  "mcpServers": {
    "my-oauth-server": {
      "url": "https://my-mcp-server.example.com",
      "oauth": {
        "clientRegistration": "cimd"
      }
    }
  }
}
```

— Multiple config keys and env vars named; one worked example `v1.0.1`

05 **searchTools() helper and codemode-deferred exposure change** 75

BREAKING

Adds searchTools() codemode helper; MCP servers with default exposure no longer appear in the system prompt description and no longer block the first prompt. The `codemode-deferred` exposure value is now an alias for the default exposure, so scripts relying on the distinct `codemode-deferred` behaviour must migrate to the default exposure and use searchTools() to access those tools.

— Migration path named but no example command `product docs`

06 **Fullscreen default TUI mode** BREAKING 75

The default TUI mode is now fullscreen; existing setups that relied on normal scrollback must set `tuiMode` to `"regular"` or pass `--tui-mode regular`.

— Exact config key and flag given for migration `product docs`

07 **Namespace grouping and guidance in codemode** NEW 65

Extensions can define a `namespace` object (with `name`, `description`, and `instructions` fields) to group related tools under one heading in codemode, mirroring how MCP servers organize tools; the `instructions` text is readable via the describeNamespace(name) hook. `pi mcp add` gained a `--description` flag, and the describeNamespace() codemode helper now returns a namespace's instructions and tool names.

— Flag and helper named, but no end-to-end example `product docs`

08 **Community extension packages added to packages page** NEW 65

New community packages listed: `@thurstonsand/pi-permissions` for user-, project-, and package-level tool permission hooks; `pi-sessions` with session search, ask, handoff, messaging, auto-titling, and indexing tools; `@diegopetrucci/pi-extensions`, a collection covering annotation UIs, context management, workflow audits, contrarian review, review-comment triage, notifications, push alerts, safety guards, GitHub research, repo-local knowledge, todos, tool rendering, and model support; `@diegopetrucci/pi-fast` to enable Fast mode for OpenAI API, OpenAI Codex, and Anthropic Claude models; `@indexyz/pi-custom-provider`, a generic provider extension supporting OpenAI-compatible, Anthropic-compatible, OpenAI Responses, and Ollama chat endpoints; `@indexyz/pi-skill` for Codex-style `$skill-name` invocation anywhere in a prompt; `@braintrust/pi-extension` for automatic tracing of sessions, turns, LLM calls, and tool executions to Braintrust; `@mutmutco/pi-plugin` for MMI workflow skills and org-gates delivery; and `pi-prompt-template-model` for a prompt template model selector in the coding agent.

Capture full session and LLM-call traces in Braintrust for observability and evaluation.

```
$ pi install npm:@braintrust/pi-extension
```

— Nine packages named with capabilities, one install command `product docs`

09 **Nix flake installation** NEW 65

Adds `nix run github:earendil-works/pi/stable` and `nix profile add github:earendil-works/pi/stable` for Nix-based install on macOS and Linux.

Install pi on a Nix-enabled system without npm or manual binary management.

```
$ nix profile add github:earendil-works/pi/stable
```

Install pi system-wide on NixOS or macOS with a pinned stable release.

```
$ nix profile add github:earendil-works/pi/stable
```

— Exact install commands given `v1.0.1`

10 **pi.registerToolRenderer for custom tool rendering** NEW 60

pi.registerToolRenderer((toolName, next) => renderers) lets extensions register `renderCall` and `renderResult` renderers for any tool — including MCP tools not yet connected in a resumed session — with next() enabling fallback chaining to another renderer.

– API signature given but no usage example v1.0.1

THINNER COVERAGE BELOW

11 **Copy sign-in URL hotkey on OAuth screens** NEW 55

Adds `app.message.copy` key (default `ctrl+x`) on OAuth sign-in screens in `/login`, `/mcp`, and `/mcp login` to copy the sign-in URL when the browser cannot be opened.

– Exact key binding and affected screens named v1.0.1

12 **Codemode prompt token reduction and error recovery hints** IMPROVED 50

Reduces codemode prompt tokens by ~40% (a GPT-5.6 request shrinks from ~5,300 to ~3,300 tokens) and adds recovery hints to codemode errors.

– Concrete numbers given but nothing to act on directly product docs

13 **MCP OAuth credentials keyed by server name and URL** BREAKING 50

MCP OAuth credentials are now stored per server name and URL; credentials previously stored by URL alone are migrated to the first server that uses that URL, which may cause unexpected account associations.

– Behavior change explained but no remediation command product docs

14 **npm-shrinkwrap.json removed from npm package** BREAKING 50

npm installations no longer include `npm-shrinkwrap.json`, so transitive dependencies are no longer pinned; use the pi.dev installer for pinned installations.

– Alternative installer named as remediation v1.0.1

15 **Cloudflare Clef classifier models** NEW 40

Adds `@cf/cloudflare/clef` and `@cf/cloudflare/clef-flash` Cloudflare Clef classifier models, usable from codemode scripts and extensions.

– Named models only, no usage example v1.0.1

└──▷ **BREAKING ON UPGRADE**

! The default TUI mode is now fullscreen; existing setups that relied on normal scrollback must set `tuiMode` to `"regular"` or pass `--tui-mode regular`.

! MCP OAuth credentials are now stored per server name and URL; credentials previously stored by URL alone are migrated to the first server that uses that URL, which may cause unexpected account associations.

! npm installations no longer include `npm-shrinkwrap.json`, so transitive dependencies are no longer pinned — use the pi.dev installer for pinned installations.

! `codemode-deferred` exposure value is now an alias for the default exposure; scripts relying on the distinct `codemode-deferred` behaviour should migrate to the default exposure and use searchTools() to access those tools.

! `npm-shrinkwrap.json` is removed from the published npm package; npm installations no longer pin transitive dependencies. Use the pi.dev installer for pinned installations.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.289 NOTES ↗

CODE ↗ RANK

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

> Claude Code expanded its agent API with new teammate-spawning capabilities, stable agent identity across plugin hooks, and richer status reporting.

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

01 **Agent spawning and status API additions** NEW 65

Adds `agent.spawn` for creating teammates, a stable agent id maintained across plugin hook events, and new idle and waiting states surfaced in $.agent.list().

— Names exact API surfaces but lacks usage detail or mechanism. v2.1.289

WAS THIS USEFUL? 👍 👎

Token Optimizer

Feedback ⓘ v5.13.30 NOTES ↗ RANK

Token Optimizer is a coding-agent plugin that trims wasted context, checkpoints work across sessions and compactions, and reports token spend on a live dashboard.

> Token Optimizer shipped a desktop status bar plugin for the Claude desktop app, bundling live session metrics, one-click session actions, and a toggle to hide it.

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

01 Desktop status bar plugin for Claude desktop app NEW

91

Installable via `/plugin install token-optimizer-desktop@alexgreensh-token-optimizer`, the new status bar appears above the prompt automatically on install or update with no extra configuration. It shows a line with the current session, its last checkpoint, and tokens saved in the last 30 days, offers one-click `Clean up`, `Start fresh`, and `Keep warm` actions, and adds a Clawd animated character that visualises what the current session is doing (requires Claude Code 2.1.287 or newer). The terminal status line now counts compactions in real time as they happen, and the whole bar can be hidden by setting `TOKEN_OPTIMIZER_STATUS_BAR=0` in the settings `env`.

Install the new desktop status bar plugin to get live token and context metrics inside the Claude desktop app.

```
$ /plugin install token-optimizer-desktop@alexgreensh-token-optimizer
```

Suppress the new status bar in environments where screen real estate or noise is a concern.

```env
TOKEN_OPTIMIZER_STATUS_BAR=0
```

— Names install command, env toggle, actions, and version requirement. v5.13.30 v5.13.29

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v10.10.0 NOTES ↗

RANK

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

> Zeroshot v10.10.0 improves run-log diagnostics, preserving node failure explanations across execution environments so retries and parallel-branch races don't erase the root cause.

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

01 **Preserved node failure explanations in run logs** IMPROVED

55

Zeroshot now preserves node failure explanations in ordinary run logs across local, Docker, and hosted environments, including cases where retries succeed or a parallel winner cancels another branch. These explanations are surfaced through the existing subscriptions and replay cursors mechanisms.

— Describes behavior and scope well but no concrete command or config shown. v10.10.0

WAS THIS USEFUL? 👍 👎

AGENT

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.17.0 NOTES ↗ RANK

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

> Gram's latest releases focus on securing and observing hosted MCP infrastructure: pinned OAuth callback URLs, new MCP server health tooling for admins, ChatGPT Desktop support, and a wave of Platform MCP methods for searching tool calls/users, managing hosted tool lists, and reviewing skill suggestions.

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

01 Pinned OAuth callback URLs for remote sessions NEW

95

Adds `GRAM_OUTBOUND_CALLBACK_URL` and `GRAM_REGISTRATION_CALLBACK_URL` environment variables to pin OAuth redirect URIs and client identity URLs so a server URL change cannot silently move existing or new remote session clients. Adds `remoteSessionClients.getNewClientCallbackUrl` to the API, returning the redirect URI a new remote session client will register; the dashboard now shows this value instead of deriving it from the server URL. Also pins MCP login IdP callbacks (`/mcp/idp_callback` , `/x/mcp/idp_callback`), the Platform MCP IdP callback, and the assistant MCP auth CIMD `client_id` and `redirect_uri` to `GRAM_OUTBOUND_CALLBACK_URL` instead of the server URL.

— Names every env var, endpoint and callback path verbatim server@2.17.0 dashboard@0.131.0

02 Platform MCP tool-call and user search NEW

90

Adds `search_tool_calls` and `list_attribute_keys` to the Platform MCP: `search_tool_calls` queries one project's tool calls across all MCP servers over up to 30 days, filtered by tool name, error text, outcome, server, person reference, and custom `@ -prefixed` attributes; `list_attribute_keys` discovers the custom and filterable system attribute keys present in that window. Also adds `search_users` (finds observed people by partial identity over up to 30 days, returning masked identities and person references) and `get_user_metrics_summary` (takes a person reference and returns call volume, failures, servers, and most-failing tools).

— Names all four methods and their filter parameters server@2.17.0

03 Platform MCP skill suggestion review workflow NEW

88

Adds `approve_skill_suggestion` and `dismiss_skill_suggestion` to the Platform MCP, enabling programmatic review of proposed skill improvements identically to the dashboard workflow. Adds an `omit_diffs` parameter to `list_skill_suggestions`, returning each suggestion's ID, rationale, and evidence counts without its diff for efficient triage of large queues. Ships a `review-skill-suggestions` workflow skill on the Platform MCP plugin that walks an administrator through reading, deciding, approving, correcting, or dismissing each queued skill improvement.

Triage a large skill-suggestion queue quickly by fetching IDs, rationales, and evidence counts without pulling full diffs for every entry.

```
$ list_skill_suggestions(project_id="<project_id>",
omit_diffs=true)
```

— Names methods, parameter and workflow skill with runnable example `server@2.17.0`

---

04 **Platform MCP tool list management on hosted servers** NEW

83

Adds `list_project_tools`, `add_tools_to_mcp`, and `remove_tools_from_mcp` to the Platform MCP, letting an agent manage the tool list on an existing hosted MCP server under confirmation, an expected version, and an idempotency key; `get_mcp` now also reports the tool list a hosted server exposes.

— Names methods and the safety parameters guarding mutation `server@2.17.0`

---

05 **ChatGPT Desktop MCP client support** NEW

75

Adds ChatGPT Desktop as a supported MCP client on the hosted install page and via `gram install chatgpt-desktop`, with Developer mode and custom connector steps matched to the server's authentication.

Install gram as an MCP server inside ChatGPT Desktop, following the generated Developer mode and custom connector steps.

```
$ gram install chatgpt-desktop
```

Install gram as an MCP server for ChatGPT Desktop, with Developer mode and custom connector configuration.

```
$ gram install chatgpt-desktop
```

Bootstrap ChatGPT Desktop as an MCP client against your gram server, including auth-matched connector setup.

```
$ gram install chatgpt-desktop
```

— Exact install command and client steps given `server@2.17.0` `dashboard@0.131.0` `cli@0.17.0`

THINNER COVERAGE BELOW

06 **Findings page on identity profiles** NEW 50

Adds a Findings page to each identity, listing the person's risk findings with filters by category and rule, with category/rule clicks on the Security tab opening pre-filtered.

— UI path described but no API or config named `dashboard@0.131.0`

WAS THIS USEFUL?  

DEPLOY

◆ Agentic Infrastructure

HeyGen HyperFrames

Feedback ⓘ v0.8.123 NOTES ↗ RANK

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

> HyperFrames added CLI and in-Studio shortcuts for faster editing access, attribution tracking for feedback, and new hooks for embedding hosts to customize Studio's timeline and thumbnail behavior.

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

01 **`useThumbnailStill` hook for embedded hosts** NEW 65

Exports `useThumbnailStill` from Studio so host applications embedding Studio can pass their own pictures through the thumbnail scheduler, ensuring previews load first.

— Names the exported hook and explains its scheduling effect v0.8.116

02 **Quick access to HyperFrames Studio** NEW 60

Adds the `hyperframes open` CLI subcommand to launch HyperFrames Studio directly from the terminal, and an Edit with Framey button inside Studio for quick access to Framey editing.

Launch HyperFrames Studio from the terminal without navigating the GUI manually.

```
$ hyperframes open
```

— Names the exact CLI subcommand with a runnable example v0.8.123

THINNER COVERAGE BELOW

03 **Author-type attribution in CLI feedback** IMPROVED 45

Feedback submitted via the CLI now records whether it was written by a person or an agent, along with the email the app attaches.

— Describes the change but no API or flag to act on v0.8.120

04 **Custom controls before timeline thumbnails toggle** NEW 35

Studio hosts can now insert their own custom controls before the timeline's thumbnails toggle.

— Names the insertion point but no API signature given v0.8.118

WAS THIS USEFUL? 👍 👎

BerriAI LiteLLM

Feedback ⓘ v1.104.0 NOTES ↗ RANK

LiteLLM is an AI gateway that calls over 100 model APIs in OpenAI format with cost tracking, guardrails, load balancing, and logging.

> LiteLLM shipped a major security hardening wave mapped to the OWASP LLM Top 10 — guardrails across every request lifecycle stage, fine-grained MCP and A2A access controls, PII/secret redaction and at-rest encryption — alongside new A2A agent-to-agent protocol support, Admin UI login lockdown and rate limiting, a raft of production scaling and Redis coordination controls, breached-password detection, and Fal.ai Seedance video generation.

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

01 Guardrails framework across the request lifecycle NEW

93

Adds a `guardrails:` config block in `config.yaml` supporting `mode: pre_call`, `mode: during_call`, or `mode: post_call` hooks that run across chat, completions, Responses API, Anthropic Messages, Google GenAI, embeddings, image generation, speech, transcription, video, rerank, OCR, Bedrock pass-through, A2A, and MCP routes. A `default_on: true` option applies a guardrail to every request without per-key or per-team attachment; built-in Content Filter patterns `prompt_injection_jailbreak`, `prompt_injection_system_prompt`, `prompt_injection_data_exfiltration`, `prompt_injection_malicious_code`, and `prompt_injection_sql` detect prompt injection, `realtime guardrails` scan Realtime session input transcription, and an opt-in Content Filter image-description path describes an image with a vision model (costing one extra vision call per request) and filters the description.

Enable prompt-injection blocking on every request before it reaches the model provider, using the built-in Content Filter patterns.

```yaml
guardrails:
  - guardrail_name: prompt-injection-block
    litellm_params:
      guardrail: content_filter
      mode: pre_call
      default_on: true
      patterns:
        - prompt_injection_jailbreak
        - prompt_injection_system_prompt
        - prompt_injection_data_exfiltration
        - prompt_injection_malicious_code
        - prompt_injection_sql
```

— Names exact config modes, routes, patterns; runnable example included product docs

02 **PII masking, secret redaction, and encryption at rest** NEW

85

Adds PII masking via Presidio, where `logging only` mode masks entities for logging callbacks while the provider sees the original text, plus secret detection that strips API keys/tokens before the provider sees them. Three redaction flags — `turn_off_message_logging`, `redact_user_api_key_info`, and `redact_messages_in_exceptions` — sit under `litellm_settings`, alongside the `store_prompts_in_spend_logs` setting (default off) controlling whether prompt/response text is written to spend logs, and the `LITELLM_SALT_KEY` environment variable encrypting model `litellm_params`, stored environment variables, and MCP credentials in the database at rest.

Encrypt secrets in the database and redact PII/keys from logs and error responses to address LLM02 Sensitive Information Disclosure.

```yaml
litellm_settings:
  turn_off_message_logging: true
  redact_user_api_key_info: true
  redact_messages_in_exceptions: true
  # store_prompts_in_spend_logs: false  # default — leave unset

# Set before first model is stored:
# export LITELLM_SALT_KEY=<your-random-secret>
```

— Exact flags and env var named with working config example product docs

03 **A2A protocol support and governance** NEW

82

Adds A2A (Agent-to-Agent) protocol support so LiteLLM can route between A2A agents alongside existing MCP tools and LLM APIs, with per-agent spend tracking shown under `Agent Usage (A2A)` in the Usage view and full request/response JSON logging when `store_prompts_in_spend_logs: true`. Production governance — SSO/SAML, audit logs, spend tracking, multi-team management, and guardrails — now extends to A2A workflows, bounded by an `A2A kill switch`, `agent_permissions`, and per-session iteration/budget caps via `max_iterations` and `max_budget_per_session`.

Lock down MCP access so keys with no explicit server list are rejected, and cap an A2A agent's iterations and spend per session.

```yaml
litellm_settings:
  require_key_mcp_access_defined: true

# On the key or team (via management API or Admin UI):
agent_permissions:
  max_iterations: 10
  max_budget_per_session: 0.50
```

Enable request/response logging for A2A agent traffic so you can inspect the full JSON payload per call.

```yaml
store_prompts_in_spend_logs: true
```

— Mechanism and UI location named; partial examples across releases `product docs`

---

04 **Production database and request load tuning** NEW 81

Adds `general_settings.proxy_batch_write_at` to batch spend writes to the database every 60 seconds instead of per-request, `general_settings.database_connection_pool_limit` to cap the per-worker DB connection pool, `general_settings.disable_error_logs: True` to stop writing LLM exceptions to the database, `litellm_settings.request_timeout` to fail hanging requests (default 6000 seconds), and `general_settings.proxy_config_reload_interval_seconds` (also via `PROXY_CONFIG_RELOAD_INTERVAL_SECONDS` or the Admin UI Router Settings) to tune how often each pod polls the database for runtime config changes when `store_model_in_db: true`.

Harden a production proxy config with batched spend writes, capped DB connections, and fast config convergence across pods.

```yaml
general_settings:
  trusted_proxy_ranges:
    - "10.0.0.0/8"
  proxy_batch_write_at: 60
  database_connection_pool_limit: 10
  disable_error_logs: True
  store_model_in_db: true
  proxy_config_reload_interval_seconds: 30

litellm_settings:
  request_timeout: 30
  set_verbose: False
  json_logs: true
```

— Five tuning keys named with runnable config example product docs

05 **MCP access control and tool-call guardrails** NEW 80

Adds `require_key_mcp_access_defined` to refuse keys carrying no MCP access list (closing default open access), plus a `no-mcp-servers` opt-out, tool allow/deny lists, per-entity tool permissions, `toolsets`, `grants`, a `client allowlist`, and `zero-trust MCP` (JWT signed with caller identity). MCP guardrails now run on every tool call via `tool filtering` to limit which tools a key can reach and a `tool permission guardrail` plus `tool policies` to allow or deny function calls by name and argument; `LITELLM_MCP_STDIO_EXTRA_COMMANDS` extends the allowlist of permitted stdio MCP server commands beyond the defaults (`npx`, `uvx`, `python`, `python3`, `node`, `docker`, `deno`).

Lock down MCP access so keys with no explicit server list are rejected, and cap an A2A agent's iterations and spend per session.

```yaml
litellm_settings:
  require_key_mcp_access_defined: true

# On the key or team (via management API or Admin UI):
agent_permissions:
  max_iterations: 10
  max_budget_per_session: 0.50
```

— Many named MCP controls but no runnable example shown product docs

06 **Redis coordination configuration for multi-worker deployments**

NEW 80

Adds `LITELLM_DISABLE_NO_REDIS_WARNING=true` to suppress the Admin UI's `no Redis` warning banner on intentional single-worker deployments, and a `general_settings.coordination_redis` config key to point proxy-wide coordination (rate limits, budgets, cache invalidation, pod lock) at a different Redis instance than the response cache. Documents that the proxy config needs both a `router_settings` block (`redis_host`, `redis_port`, `redis_password`) for router state like cooldowns, and a `litellm_settings.cache_params` block (`type: redis`) for response caching and coordination — without Redis, rate limits, budgets, cooldowns, cache invalidation, scheduled jobs, MCP outbound OAuth tokens, request prioritization, and SSO login flows all degrade or fail across multiple workers or pods.

Suppress the `no Redis` Admin UI banner when you intentionally run a single-worker dev instance.

```
$ LITELLM_DISABLE_NO_REDIS_WARNING=true litellm --config
config.yaml --num_workers 1
```

Wire up Redis fully for a multi-worker or Kubernetes deployment — configure both the router state block and the cache block so all enforcement (rate limits, budgets, cooldowns, cache hits) is shared across workers.

```yaml
router_settings:
  redis_host: os.environ/REDIS_HOST
  redis_port: os.environ/REDIS_PORT
  redis_password: os.environ/REDIS_PASSWORD

litellm_settings:
  cache: True
  cache_params:
    type: redis
    host: os.environ/REDIS_HOST
    port: os.environ/REDIS_PORT
    password: os.environ/REDIS_PASSWORD
```

Point proxy-wide coordination (pod lock, budget resets, cache invalidation) at a dedicated Redis separate from the response cache Redis.

```yaml
general_settings:
  coordination_redis:
    redis_host: os.environ/COORD_REDIS_HOST
    redis_port: os.environ/COORD_REDIS_PORT
    redis_password: os.environ/COORD_REDIS_PASSWORD

litellm_settings:
  cache: True
  cache_params:
    type: redis
    host: os.environ/CACHE_REDIS_HOST
    port: os.environ/CACHE_REDIS_PORT
    password: os.environ/CACHE_REDIS_PASSWORD
```

— Dual config blocks and three runnable examples given `product docs`

07 **Background job and worker lifecycle controls** NEW 78

Adds the `LITELLM_JOB_ROLE` environment variable (`all` , `worker` , or `serving`) to control which background jobs — budget resets, spend-log cleanup, usage exports — a pod registers, letting a dedicated worker deployment own shared jobs; `--max_requests_before_restart` recycles each Uvicorn worker after a fixed request count to bound memory growth; `--num_workers` / `NUM_WORKERS` is documented with per-topology guidance (one worker per pod on Kubernetes, vCPU count on a bare VM); and a `litellm_pod_lock_manager_size` Prometheus gauge with a `<job>:<pod>` label exposes which pod holds each background-job lock.

Pin a multi-pod deployment so one dedicated pod owns shared background jobs (budget resets, spend cleanup, usage exports) and serving pods handle only traffic — the only safe approach when Redis is not configured.

```yaml
# worker Deployment
env:
  - name: LITELLM_JOB_ROLE
    value: "worker"

# serving Deployment
env:
  - name: LITELLM_JOB_ROLE
    value: "serving"
```

Run one worker per pod on Kubernetes with automatic worker recycling to bound memory growth under sustained load.

```
$ CMD ["--port", "4000", "--config",
"./proxy_server_config.yaml", "--num_workers", "1", "--
max_requests_before_restart", "10000"]
```

— Four controls named with two runnable examples `product docs`

---

08 **File upload restrictions and size limits** NEW 67

Adds `general_settings.allowed_file_extensions` (e.g. `[".jsonl", ".pdf", ".txt"]`) to restrict `POST /v1/files` uploads by extension, returning a 400 OpenAI-shaped error for disallowed types, plus `general_settings.max_file_size_mb`, `general_settings.max_batch_file_size_mb`, and `general_settings.max_request_size_mb` to cap upload and request body sizes.

Restrict file uploads to only batch and document types, cap sizes, and prevent oversized request bodies across all routes.

yaml

```yaml
general_settings:
  master_key: sk-...
  allowed_file_extensions: [".jsonl", ".pdf", ".txt"]
  max_file_size_mb: 50
  max_batch_file_size_mb: 100
  max_request_size_mb: 10
```

— Exact keys, formats, and runnable yaml example given `product docs`

---

09 **Env credential login lockout for Admin UI** BREAKING 65

Adds `general_settings.disable_env_credential_login: true` in `config.yaml` to block `UI_USERNAME` / `UI_PASSWORD` and master-key logins to the Admin UI, forcing database or SSO accounts only. Enabling it before creating at least one `proxy_admin` database user locks all admins out of the Admin UI; the fix is to remove the setting and restart to restore environment login.

Harden a production deployment by disabling shared env-var Admin UI credentials after creating individual admin accounts.

yaml

```yaml
general_settings:
  disable_env_credential_login: true
```

– Exact flag, example, and lockout consequence documented product docs

10 **Admin UI login brute-force rate limiting** NEW 63

Adds `general_settings.trusted_proxy_ranges` (CIDR list) and `max_failed_login_attempts_per_source_overrides` (per-IP map) to tune Admin UI brute-force rate limiting, which by default blocks a source address for 5 minutes after more than 10 wrong passwords in 60 seconds; blocks are shared across workers and pods via Redis and usernames are stored hashed. The same `trusted_proxy_ranges` key also scopes this rate limiting to the real client address behind a load balancer.

– Thresholds and keys named, defaults specified, no example product docs

11 **Disable API documentation endpoints in production** NEW 60

Adds `NO_DOCS`, `NO_REDOC`, and `NO_OPENAPI` environment variables (set to `"True"`) to independently disable Swagger UI, ReDoc, and `/openapi.json` in production.

Remove all API documentation surfaces from a production proxy to reduce reconnaissance exposure.

```
$ NO_DOCS="True" NO_REDOC="True" NO_OPENAPI="True" litellm --config config.yaml
```

– Exact env vars and runnable shell command given product docs

THINNER COVERAGE BELOW

12 **Structured logging and Slack alerting** NEW 53

Documents `json_logs: true` under `litellm_settings` and `LITELLM_LOG="ERROR"` for structured production logging with debug output suppressed, and Slack alerting via `alerting: ["slack"]` under `general_settings` plus the `SLACK_WEBHOOK_URL` environment variable, configurable in the UI under Settings > Logging & Alerts > Alerting Types.

– Keys and UI path named, no dedicated example product docs

13 **Secure cookies behind a reverse proxy** NEW 52

Adds the `PROXY_BASE_URL` environment variable (set to the public `https://` origin), `general_settings.use_x_forwarded_for: true`, and `general_settings.mcp_trusted_proxy_ranges` to ensure session, SSO, and SAML cookies are marked Secure when TLS terminates at a reverse proxy.

– Three settings named with clear purpose, no example product docs

14 **Master key startup safety checks** BREAKING 52

Documents `LITELLM_MASTER_KEY` (must start with `sk-`) as the proxy admin credential, and the proxy now refuses to start if `LITELLM_MASTER_KEY` is unset, empty, or a publicly known value such as `sk-1234` — deployments using the default key will fail to boot until a strong secret is set.

— Exact env var and failure condition, no example v1.104.0

15 **Virtual key and request access restrictions** NEW 48

Adds `allowed_routes` on virtual keys to restrict which proxy routes a key can reach, `model_info.access_groups` plus model access groups on keys/teams to grant named sets of models instead of raw model names, and `api_base` / `base_url` request-body blocking so callers cannot redirect the gateway to an arbitrary upstream.

— Config keys named but thinly described, no example product docs

16 **Supply-chain hardened container images** NEW 40

Ships cosign-signed container images built from a Chainguard Wolfi base pinned by digest, a non-root image variant, a frozen `uv.lock` lockfile, Grype/OSV CI scanning, and OpenSSF Scorecard reporting.

— Lists hardening techniques but no actionable command product docs

17 **Fal.ai Seedance video generation** NEW 40

Adds Seedance 2.5 and 2.0 video generation support via the fal.ai queue API.

— Named models and API, no example or detail v1.104.0

18 **Per-tenant response cache namespacing** NEW 38

Adds a `cache.namespace` config key and a `metadata.redis_namespace` request field to scope the exact-match response cache per tenant.

— Two fields named, minimal mechanism, no example product docs

19 **Breached password detection and forced resets** NEW 36

Adds breached password detection and self-service password change to the proxy authentication password policy, forcing a password reset when a password is found breached or was set by an admin, and closes plaintext password leaks in auth flows.

— Described only in prose, no config name given v1.104.0

20 **SCIM-based key deprovisioning** NEW 27

Supports SCIM provisioning to automatically remove keys and access tokens when a user is deprovisioned from the identity provider.

— Thin prose description, no config name or example `product docs`

└──▷ **BREAKING ON UPGRADE**

! Setting `general_settings.disable_env_credential_login: true` before creating at least one `proxy_admin` database user will lock all admins out of the Admin UI; remove the setting and restart to restore environment login.

! The proxy now refuses to start if `LITELLM_MASTER_KEY` is unset, empty, or a publicly known value — deployments using the default `sk-1234` key will fail to boot and must set a strong secret before upgrading.

WAS THIS USEFUL?  

ArcBox

Feedback ⓘ v0.8.2 NOTES ↗ RANK

ArcBox runs containers, microVM agent sandboxes, and full Linux or macOS virtual machines on a Mac from one Rust daemon and CLI.

> ArcBox's v0.8.2 release adds full machine lifecycle management (clone, export/import, resize, default) plus NFSv3 root filesystem exports and mDNS-based local domain resolution for Linux VMs, while fleet-agent v0.1.7 trims attach latency by completing host facts right after handshake.

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

01 **Machine lifecycle subcommands: clone, export, import, resize**

NEW 88

Adds `machine clone`, `machine export`, `machine import`, `machine resize`, and `machine default` subcommands to `abctl`, backed by matching `clone`, `export`, `import`, `resize`, and `default` RPCs added to the machine protocol and API. Export and import move machines through a sparse `tar.zst` archive format, clone produces an independent copy-on-write environment from a stopped machine, and resize applies live CPU and memory changes to an already-running machine.

Clone a stopped machine to spin up an identical copy-on-write environment for testing without touching the original.

```
$ abctl machine clone dev dev-clone
```

Export a machine to a portable archive, then import it on another host or keep it as a snapshot.

```
$ abctl machine export dev dev-backup.tar.zst
abctl machine import dev-restored dev-backup.tar.zst
```

— Names every subcommand, RPC and archive format with runnable examples v0.8.2

02 **NFSv3 root filesystem export for machines** NEW 74

Serves each running machine's root filesystem as an NFSv3 export, mounted under `~/ArcBoxMachines` on the host, driven by the new `EnsureMachineExport` agent RPC over the bridge NIC. Docker export and machine roots are now mounted under a single unified host mount root.

— Names RPC and mount path but gives no command to try it v0.8.2

THINNER COVERAGE BELOW

03 **Automatic default machine targeting for exec and ssh** IMPROVED

45

Extends `machine exec` and `machine ssh` to pick the default machine and a target user automatically, removing the need to specify them explicitly.

— Names the two commands but no concrete example or flag v0.8.2

04 **Faster fleet agent attach via immediate host facts** IMPROVED

35

Completes host fact collection immediately after the handshake rather than deferring it, reducing attach latency for fleet-managed workloads in fleet-agent v0.1.7.

— Explains the change but no numbers or config surface fleet-agent-v0.1.7

05 **mDNS local-domain resolution for machines** NEW

27

Registers machine local-domain names with mDNSResponder for host-side DNS resolution.

— Bare description with no mechanism or usage detail v0.8.2

WAS THIS USEFUL? 👍 👎

Modal

Feedback ⓘ 1.6.1 (2026-10-03) NOTES ↗ RANK

Bring your own code, and run CPU, GPU, and data-intensive compute at scale. The serverless platform for AI and data teams.

> Modal shipped new CLI visibility into Endpoints, affinity-based request routing, and finer-grained container metrics across its SDK and session API.

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

01 **Endpoint observability CLI commands** NEW 80

Adds `modal endpoint info` to display basic metadata about Modal Endpoints, `modal endpoint stats` to report aggregate performance metrics (container utilization, throughput), and `modal endpoint logs` to fetch or stream logs from deployed Endpoints for real-time production debugging.

Stream live logs from a deployed Modal Endpoint to debug production traffic in real time.

```
$ modal endpoint logs
```

Check aggregate performance metrics for a Modal Endpoint, including container utilization.

```
$ modal endpoint stats
```

Stream live logs from a specific Modal Endpoint to debug latency or errors in production.

```
$ modal endpoint logs <endpoint-name>
```

Check aggregate performance metrics for an Endpoint to understand throughput and concurrency.

```
$ modal endpoint stats <endpoint-name>
```

— Three runnable CLI commands with examples, each explained 1.6.1 (2026-10-03)

02 **Container count metric in stats** NEW 75

Adds `container_total_count` to `modal function stats`, `modal server stats`, Function.stats(), and Server.stats(), reporting the number of containers live at

any point in the interval.

— Names exact field across four reporting surfaces `1.6.1 (2026-10-03)`

---

03 **Affinity routing via request header** NEW

65

Adds affinity routing support via the `Modal-Routing-Affinity-Key` HTTP request header, allowing clients to be consistently routed to the same backend instance by supplying a client-provided key.

— Names exact header but no usage example given `product docs`

---

04 **Longer wait for session container placement** IMPROVED

60

Extends Server.sessions.start() to wait up to 25 minutes when no container is available to place the session.

— Names method and limit, no further behavioural detail `1.6.1 (2026-10-03)`

WAS THIS USEFUL?  

Google Gemini Enterprise Agent Platform

Feedback ⓘ changelog · Oct 4, 2026 CHANGELOG ↗ RANK

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

> Gemini Enterprise Agent Platform shipped a broad release spanning governance controls (custom denial messages, VPC Service Controls), new observability (Cloud Trace, GA App Topology API, a Topologies query builder), new model support (Gemini 3, Gemini 3.8 Flash TTS, Grok 4.7, Claude Sonnet 5.5), Provisioned Throughput order changes, and a cluster of CodeMender hardening and reliability fixes including a breaking location-config change.

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

01 Custom denial messages for governance policies NEW

85

Adds `agentResponseCustomization.denialMessage` field on semantic governance policy resources, settable via REST API or the Google Cloud console, to show a custom fixed message of up to 1,000 characters when a policy denies a request.

Configure a custom denial message on a semantic governance policy so end users see a clear, actionable message instead of the raw policy rationale.

```json
{
  "agentResponseCustomization": {
    "denialMessage": "This request was blocked by your organization governance policy. Contact your admin for access."
  }
}
```

— Named field, character limit, config example and setting method

changelog-20261004-a78641bb

02 Gemini 3.8 Flash TTS models on global endpoint NEW

80

Adds Gemini 3.8 Flash TTS and Gemini 3.8 Flash-Lite TTS models (Preview) on the global endpoint, supporting structured speaker and style metadata, inline vocal tags, multi-speaker dialogue, streaming, voice design, and voice replication.

— Rich capability list but no API call shown changelog-20261004-a78641bb

03 **CI gate severity validation in CodeMender** NEW 80

Adds CI gate severity validation in CodeMender (v0.11.0), rejecting misspelled `CRITICAL`, `HIGH`, `MEDIUM`, `LOW`, or `NONE` severity values up front rather than silently ignoring them.

— Names exact severity values, behavior and version changelog-20261004-a78641bb

04 **Hardened VCS branch handling in CodeMender architecture sessions**

IMPROVED 80

Enables `vcs_branch` creation and candidate branch cleanup inside the default sandboxed worker for `--architecture` sessions, and hardens VCS template expansion against shell metacharacters and command injection.

— Names flag and security hardening mechanism changelog-20261004-a78641bb

05 **CM_LOCATION replaces --location flag** BREAKING 75

CodeMender now sets service location via the `CM_LOCATION` environment variable and `location` config key, replacing the `--location` command-line flag; existing scripts using `--location` will need to be updated.

Set the CodeMender service location via environment variable instead of the removed `--location` flag, e.g. in a CI pipeline step.

```
$ export CM_LOCATION=us-central1
```

— Exact env var/config key named with a runnable replacement example

changelog-20261004-a78641bb

06 **Retry backoff and stream reconnection under rate limiting**

IMPROVED 70

Extends HTTP 429 retry backoff with Retry-After support and up to 60-second quota refill windows, plus automatic stream reconnection for scan resilience under rate limiting.

— Names header and limit but no config exposed changelog-20261004-a78641bb

07 **Cloud Trace integration with Agent Gateway** NEW 60

Adds Cloud Trace integration with Agent Gateway (Preview) for end-to-end request observability across agents, tools, MCP servers, and Google Cloud services.

— Names integration scope but no setup steps changelog-20261004-a78641bb

08 **VPC Service Controls for governance policy engine** NEW

60

Supports VPC Service Controls for semantic governance policies and the policy engine (Preview), enabling inclusion in a service perimeter to mitigate data-exfiltration risk.

— Explains purpose but lacks setup detail changelog-20261004-a78641bb

09 **Session resume on interrupt in CodeMender** IMPROVED

60

Preserves standalone sessions on `SIGINT` / `SIGTERM` interrupt so interrupted runs can be resumed, while still canceling parallel consensus worker operations.

— Names signals and behavior but no resume command shown changelog-20261004-a78641bb

THINNER COVERAGE BELOW

10 **Query builder on Topologies page** NEW

55

Adds a query builder on the Topologies page for writing custom queries against agent identity, alerts, vulnerabilities, and underlying infrastructure.

— UI path given but no query syntax detail changelog-20261004-a78641bb

11 **Single-agent topology view in Agent Registry** NEW

50

Adds a streamlined single-agent topology view in Agent Registry that auto-loads single-hop traffic topology.

— UI feature named but little further detail changelog-20261004-a78641bb

12 **New model support across Interactions API and catalog** NEW

45

Adds Gemini 3 model support to the Interactions API (Preview); adds xAI Grok 4.7 model support; and adds Anthropic Claude Sonnet 5.5 model support.

— Names models but no usage mechanism changelog-20261004-a78641bb

13 **Supersede-and-replace for Provisioned Throughput orders** NEW

45

Adds support for changing order scope and increasing term via a supersede-and-replace operation in Provisioned Throughput.

— Names operation but lacks a command example changelog-20261004-a78641bb

14 **App Topology API reaches GA** IMPROVED

35

Promotes the App Topology API, which provides agent topologies in Gemini Enterprise Agent Platform, to GA.

— Bare GA announcement with little mechanism changelog-20261004-a78641bb

└──▷ BREAKING ON UPGRADE

! The `--location` command-line flag in CodeMender is replaced by the `CM_LOCATION` environment variable and `location` config key; existing scripts using `--location` will need to be updated.

WAS THIS USEFUL?  

DATA

◆ Vector DBs & RAG

GrowGraph OntoCast

Feedback ⓘ v0.6.6 CODE ↗ RANK

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

> OntoCast's documentation window exposes a vastly expanded configuration surface — vector-based ontology retrieval with induced-subgraph and fusion tuning, SHACL facts validation with an autofix/critic loop, entity-aggregation guards, web-search grounding, and chunking/conversion controls — alongside new HTTP endpoints for catalog and per-unit processing, a config-auditing CLI, and breaking default-model and conversion-profile changes in v0.6.5/v0.6.6.

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

01 **LLM provider configuration: model, credentials, caching, rate limits, reasoning**

NEW 97

`LLM_PROVIDER` selects the backend (`openai` , `ollama` , `anthropic` , `google` , each needing its own install extra); `LLM_API_KEY` is the single credential variable for OpenAI/Anthropic/Google, and `LLM_BASE_URL` points at a custom endpoint (required for `ollama` , usable for any OpenAI-compatible service). `LLM_MODEL_NAME` (default `gpt-5.6-luna`) passes any provider-accepted model name through, with presets added for GPT-5.6 (`gpt-5.6` , `gpt-5.6-sol` , `gpt-5.6-terra` , `gpt-5.6-luna`), GPT-6 (`gpt-6-astra` , `gpt-6-sol` , `gpt-6-luna`), Gemini 3.x, and current Ollama models. `LLM_TEMPERATURE` defaults to `0.0` (OpenAI `gpt-5` / `gpt-5-mini` / `gpt-5-nano` pinned to `1.0`). The response cache uses `LLM_CACHE_ENABLED` , `LLM_CACHE_READ_ONLY` , and `LLM_PROMPT_CACHE_KEY` (OpenAI only). Throughput is bounded by `LLM_MAX_INFLIGHT` (alias `MAX_INFLIGHT`), `LLM_REQUEST_TIMEOUT_SECONDS` , `LLM_REQUESTS_PER_SECOND` , and `LLM_MAX_RETRIES` (ignored by Ollama). Ollama reasoning controls are `LLM_THINK` , `LLM_NUM_PREDICT` , and `LLM_NUM_CTX` . `LLM_JSON_MODE` (OpenAI only) constrains decoding to `response_format json_object` . `LLM_REASONING_EFFORT` (`none` to `max`) is sent as `reasoning_effort` to OpenAI reasoning models and `thinking_level` to Gemini 3+, and `LLM_THINKING_BUDGET` sets Gemini 2.5's thinking-token budget (`0` disables, `-1` lets the model choose); both form part of the LLM cache key and cannot be combined.

Run OntoCast against a local Ollama thinking model, giving it enough context and output tokens for deep reasoning.

```
$ export LLM_PROVIDER=ollama
export LLM_BASE_URL=http://localhost:11434
export LLM_MODEL_NAME=qwen3
export LLM_THINK=true
export LLM_NUM_CTX=32768
export LLM_NUM_PREDICT=8192
export LLM_MAX_INFLIGHT=4
```

Throttle an OpenAI deployment to stay inside a rate-limited provider tier, capping both concurrency and sustained request rate while enabling JSON-safe responses.

```
$ export LLM_PROVIDER=openai
export LLM_API_KEY=sk-...
export LLM_MODEL_NAME=gpt-5-mini
export LLM_MAX_INFLIGHT=8
export LLM_REQUESTS_PER_SECOND=2.0
export LLM_REQUEST_TIMEOUT_SECONDS=30
export LLM_MAX_RETRIES=5
export LLM_JSON_MODE=true
export LLM_PROMPT_CACHE_KEY=prod-deployment-1
```

Enable read-only cache mode during a CI replay run so existing cached LLM responses are used but no new entries are written, keeping the cache stable.

```
$ export LLM_CACHE_ENABLED=true
export LLM_CACHE_READ_ONLY=true
```

Avoid hitting a provider's requests-per-minute ceiling by pacing calls just below the limit and capping concurrency.

```
$ export LLM_REQUESTS_PER_SECOND=8
export LLM_MAX_INFLIGHT=5
export LLM_MAX_RETRIES=4
ontocast process --input-path doc.pdf --output-dir ./out
```

— 17 named vars plus new presets, each with runnable examples. v0.6.6

02 **SHACL facts validation, critic loop, and literal cleanup** NEW

`FACTS_SHAPES_DIR` loads a directory of SHACL shape files (`.ttl` , recursive) into the tenant's shapes partition at startup (requires the `shacl` extra; without it validation logs a warning and does nothing). `FACTS_SHACL_INFERENCE` (`none` | `rdfs` | `owlrl` , default `rdfs`) sets pyshacl inference level; `FACTS_SHACL_ADVANCED` (default `true`) enables SHACL Advanced Features like `sh:sparql` ; `FACTS_SHACL_AUTOFIX` (`off` | `rewrite` | `prune` , default `prune`) does LLM-free repair bounded by `FACTS_SHACL_AUTOFIX_PASSES` in a validate→autofix→revalidate loop that reverts non-improving passes; `FACTS_SHACL_MAX_TRIPLES` skips validation with a visible warning on oversized graphs. `FACTS_SHAPES_PROMPT_CONTRACT` (`off` | `auto` | `full` | `context` , default `auto`) and `FACTS_SHAPES_PROMPT_MAX_LINES` surface shapes to the renderer/critic as a prompt conformance chapter. `FACTS_OBJECT_PROPERTY_LITERAL_CHECK` quarantines string literals on class-range predicates for critic IRI resolution. The critic loop is tuned by `FACTS_CRITIC_MAX_DELETE_SHARE` , `FACTS_CRITIC_ALLOW_SUBJECT_RENAME` (default `false`), `FACTS_CRITIC_MIN_TRIPLES` , and `FACTS_ACCEPT_BLOCKING_SEVERITY` (`critical` | `important` | `never` , default `critical`); `FACTS_PROPERTY_ALIAS_MIN_RATIO` floors the near-miss property rewrite and `FACTS_MERGE_REPAIR_PASSES` bounds the post-aggregation un-merge budget (`0` records findings without repairing). Data-quality knobs include `FACTS_NUMERIC_IDENTIFIER_GUARD` , `FACTS_CONTEXT_FROM_UNITS` , `FACTS_SUSPECT_MULTI_VALUE_SEVERITY` (`error` | `warning`) and `FACTS_SUSPECT_MULTI_VALUE_REQUIRE_CROSS_UNIT` , `FACTS_DOMAIN_ADHERENCE_MIN_SHARE` / `FACTS_DOMAIN_ADHERENCE_MIN_TERMS` , `FACTS_ADDITIONAL_STANDARD_NAMESPACES` (default includes `https://schema.org/`), `FACTS_QUANTITY_FALLBACK_VOCABULARY` (defaults to QUDT), `FACTS_LITERAL_VARIANT_DEDUPE` , `FACTS_NUMERIC_COVERAGE_LIMIT` and `FACTS_NUMERIC_COVERAGE_MANDATORY` (`off` | `measurements` | `all`), `FACTS_COMPLETION_PASSES` , and `FACTS_CODE_PREDICATES` (default includes `qudt:ucumCode` , `qudt:symbol`).

Enable SHACL validation against custom shape files with OWL-RL inference and auto-repair, so violations are caught and pruned before acceptance.

```ini
FACTS_SHAPES_DIR=/opt/ontocast/shapes
FACTS_SHACL_INFERENCE=owlrl
FACTS_SHACL_AUTOFIX=prune
FACTS_SHACL_AUTOFIX_PASSES=3
FACTS_SHACL_MAX_TRIPLES=500000
```

Show SHACL shape rules to the LLM critic only for shapes relevant to the current unit's ontology snapshot, keeping prompt size under control for large catalogs.

```ini
FACTS_SHAPES_PROMPT_CONTRACT=context
FACTS_SHAPES_PROMPT_MAX_LINES=200
```

Tune the critic loop to run two review passes, skip empty renders, block only on critical findings, and require cross-unit evidence before flagging multi-valued IRI predicates as errors.

```ini
FACTS_CRITIC_PASSES=2
FACTS_CRITIC_MIN_TRIPLES=1
FACTS_ACCEPT_BLOCKING_SEVERITY=critical
FACTS_SUSPECT_MULTI_VALUE_REQUIRE_CROSS_UNIT=true
FACTS_CRITIC_MAX_DELETE_SHARE=0.25
FACTS_CRITIC_MIN_DELETES=2
```

— Enumerates ~28 vars with defaults, mechanisms, and the gating extra. product docs

03 Pipeline orchestration and prompt-budget configuration NEW

96

`RENDER_MODE` (`ontology` | `facts` | `ontology_and_facts`) selects what the pipeline renders. `ONTOLOGY_CHAPTER_FORMAT` (`auto` | `inherit` | `turtle` | `term_sheet`) sets the ontology-chapter syntax in facts prompts, and `ONTOLOGY_CONTEXT_SCOPE` (`unit` | `document`) resolves it per unit or per document, paired with `FANOUT_WARMUP_UNITS` to seed a provider prefix cache before fan-out. Prompt size is bounded by `ONTOLOGY_CONTEXT_MAX_TRIPLES` , `ONTOLOGY_MAX_TRIPLES` (runaway-growth backstop), `ONTOLOGY_TEXT_MAX_CHARS_NAMING` / `_CONTRACT` / `_PROSE` , and `ONTOLOGY_TEXT_TOTAL_BUDGET` (reported via `budget.counters` fields `chapter/text_chars_before` , `chapter/text_chars_after` , `chapter/literals_clipped` , `chapter/text_over_budget`); `ONTOLOGY_CONTEXT_REQUIRED` fails a run when a facts unit's ontology context is empty. `ENABLE_ONTOLOGY_CONSOLIDATION` runs an optional consolidation pass after normalization. Concurrency/service controls include `PARALLEL_WORKERS` , `MAX_CONCURRENT_PROCESSES` (queues excess `/process` / `/process_unit` requests), `MAX_TENANCY_SCOPES` (LRU-evicts resident tenant/project ToolBoxes), `HOST` / `PORT` (default `127.0.0.1:8999`), `MAX_VISITS_PER_NODE` , `LOGGING_LEVEL` , `CLEAN` (flush triple store before loading ontologies in batch mode), and `CURRENT_DOMAIN` (base IRI stem, `<domain>/doc/<hash>`).

Extract facts only against a curated catalog pinned by ID, failing fast if the ontology context is missing, with prompt-friendly Turtle chapter format.

```ini
RENDER_MODE=facts
ONTOLOGY_CONTEXT_MODE=fixed_single_ontology
ONTOLOGY_CONTEXT_FIXED_ONTOLOGY_ID=https://example.org/ontologies
/my-catalog
ONTOLOGY_CHAPTER_FORMAT=term_sheet
ONTOLOGY_CONTEXT_REQUIRED=true
LLM_GRAPH_FORMAT=jsonld
```

Cap ontology chapter size to control prompt length, then warm up prefix-cache sharing across concurrent units in a large document.

```ini
ONTOLOGY_CONTEXT_SCOPE=document
ONTOLOGY_CONTEXT_MAX_TRIPLES=500
ONTOLOGY_TEXT_TOTAL_BUDGET=8000
FANOUT_WARMUP_UNITS=2
PARALLEL_WORKERS=8
```

Run a fresh batch load in a container, binding the server to all interfaces and clearing stale triples before ingestion.

```ini
HOST=0.0.0.0
PORT=8080
CLEAN=true
LOGGING_LEVEL=info
```

Lock fact extraction to a single settled ontology and abort rather than fall back to generic vocabulary if the catalog is missing or mismatched.

```ini
RENDER_MODE=facts
ONTOLOGY_CONTEXT_MODE=fixed_single_ontology
ONTOLOGY_CONTEXT_FIXED_ONTOLOGY_ID=my_schema
ONTOLOGY_CONTEXT_REQUIRED=true
```

Run facts-only extraction against a fixed ontology using vector retrieval, persisting results to Fuseki, while failing fast if the catalog is empty.

```ini
RENDER_MODE=facts
ONTOLOGY_CONTEXT_MODE=selected_vector_search_ontology
ONTOLOGY_CONTEXT_REQUIRED=true
FUSEKI_URI=http://localhost:3030/myds
FUSEKI_AUTH={"user":"admin","password":"secret"}
LLM_PROVIDER=openai
LLM_MODEL_NAME=gpt-5.6-luna
LLM_API_KEY=sk-...
PARALLEL_WORKERS=4
LLM_MAX_INFLIGHT=8
```

— 20 named config keys with defaults and runnable examples. product docs

04 **Vector-based ontology retrieval and induced-subgraph tuning**

NEW 95

`VECTOR_STORE_BACKEND` (`auto` | `qdrant` | `lancedb` | `none`) selects the vector-store implementation, inferred from `QDRANT_URI` / `LANCEDB_ENABLED` (plus `LANCEDB_DATA_DIR` , `QDRANT_API_KEY`) with mutual exclusion enforced at startup; setting `ONTOLOGY_CONTEXT_MODE=selected_vector_search_ontology` turns on vector-based retrieval per content unit. Embeddings use `EMBEDDING_PROVIDER` (Hugging Face, `openai` , or `ollama`), `EMBEDDING_MODEL_NAME` , `EMBEDDING_DIMENSION` , `EMBEDDING_QUERY_PREFIX` / `EMBEDDING_DOCUMENT_PREFIX` , and `VECTOR_STORE_EMBEDDING_BATCH_SIZE` . Retrieval depth and fusion are tuned by `VECTOR_STORE_TOP_K` , `VECTOR_STORE_BM25_TOP_K` , `VECTOR_STORE_FUSION_CORE_WEIGHT` / `VECTOR_STORE_FUSION_NEIGHBORHOOD_WEIGHT` / `VECTOR_STORE_FUSION_BM25_WEIGHT` , `VECTOR_STORE_FUSION_RANK_CONSTANT` , and `VECTOR_STORE_CONSISTENCY_CRITIC_MIN_FUSED_SCORE` . Induced-subgraph expansion uses `VECTOR_STORE_INDUCED_SUBGRAPH_DEPTH` , `_HUB_SEED_COUNT` , `_ANCESTOR_CLOSURE_DEPTH` , `_MAX_TOTAL_TRIPLES` , `_ESTIMATED_TRIPLES_PER_QUERY` , `_TYPE_PROMOTION_SCORE_FACTOR` , `_SEED_ORDER` (`score` | `ontology_round_robin`), `_SYMBOL_PREDICATES` , and `_CANDIDATE_PUSHDOWN` ; proposition windowing uses `VECTOR_STORE_PROPOSITION_WINDOW_SENTENCES` / `_STRIDE` / `_MAX_CHARS` / `_MAX_TOKENS` / `_OVERLAP` , `_ABBREVIATION_AWARE` , `_MEASUREMENT_AWARE` , `_MAX_WINDOWS` , `_RETRIEVAL_ENABLED` . Index lifecycle uses `VECTOR_STORE_REINDEX_CONCURRENCY` , `VECTOR_STORE_WIPE_ON_INIT` (also via `--wipe-vector-store` CLI flag), `VECTOR_STORE_PRUNE_ORPHAN_IRIS_ON_INIT` , `VECTOR_STORE_MINIMAL_LABEL_LIMIT` , `VECTOR_STORE_INDEX_UNDESCRIBED_IRIS` , `VECTOR_STORE_EMBED_STANDARD_VOCAB_IRIS` ,

`VECTOR_STORE_EXTRA_EXCLUDED_NAMESPACE_PREFIXES`, and `VECTOR_STORE_DEDUP_MODE` (`iri` | `atom_id`) with `VECTOR_STORE_DEDUP_INCLUDE_VERSION`. Ontology-patch retrieval volume is capped by `ONTOLOGY_PATCH_MAX_ATOMS`, `ONTOLOGY_PATCH_MAX_ATOMS_BASE`, and `ONTOLOGY_PATCH_SMALL_MODULE_CLOSURE_MAX_TOTAL_TRIPLES`, with `ONTOLOGY_PATCH_DUMP_ONTOLOGY_RANKS=true` exposing diagnostics in `patch_retrieval.ontology_rank_diagnostics`. A `ToolBox.ensure_vector_store` method prepares a store on demand; `metadata.retrieval_metrics` in `/process` responses reports `ontology_context_mode`, `ontology_snapshot_triples`, `empty_snapshot_reason`, and `patch_retrieval.seeds_by_ontology`; a `llm/ontology_selection` counter tracks per-unit ontology-selection calls; and in vector mode a consistency critic surfaces suggestions about new terms resembling existing catalog terms via `metadata.improvement_suggestions`.

Switch to LanceDB as the vector backend and tune the induced-subgraph triple budget before starting a ToolBox session — set these in your environment or .env file before invoking Python.

```ini
VECTOR_STORE_BACKEND=lancedb
VECTOR_STORE_TOP_K=20
VECTOR_STORE_INDUCED_SUBGRAPH_MAX_TOTAL_TRIPLES=4000
VECTOR_STORE_INDUCED_SUBGRAPH_SEED_ORDER=ontology_round_robin
```

Enable token-aware proposition windowing with overlap to avoid splitting measurement expressions across query windows — no reindex needed, applies at query time.

```ini
VECTOR_STORE_PROPOSITION_WINDOW_MAX_TOKENS=256
VECTOR_STORE_PROPOSITION_WINDOW_OVERLAP=0.1
VECTOR_STORE_PROPOSITION_MEASUREMENT_AWARE=true
VECTOR_STORE_PROPOSITION_ABBREVIATION_AWARE=true
```

Perform a clean-slate reindex after swapping embedding models, pruning orphaned IRIs and excluding an upper ontology namespace from competing in retrieval.

```ini
VECTOR_STORE_WIPE_ON_INIT=true
VECTOR_STORE_PRUNE_ORPHAN_IRIS_ON_INIT=true
VECTOR_STORE_EXTRA_EXCLUDED_NAMESPACE_PREFIXES=
["http://purl.obolibrary.org/obo/BFO_",
"http://www.w3.org/ns/sosa/"]
```

Start the OntoCast server in vector-retrieval mode with LanceDB so each content unit is matched against the most relevant terms from a large multi-topic ontology catalog.

```
$ pip install "ontocast[server,openai,doc-processing,lancedb]"
export LANCEDB_ENABLED=true
export ONTOLOGY_CONTEXT_MODE=selected_vector_search_ontology
ontocast serve --ontology-dir ./my-ontologies
```

Rebuild the vector partition after swapping the embedding model, then restart in vector mode so the new embeddings take effect.

```
$ export ONTOLOGY_CONTEXT_MODE=selected_vector_search_ontology
ontocast serve --wipe-vector-store --ontology-dir ./my-ontologies
```

Enable vector-based ontology context retrieval so each content unit is matched against the most relevant ontology rather than a single working one.

```
$ ONTOLOGY_CONTEXT_MODE=selected_vector_search_ontology ontocast
extract ./docs/
```

— Enumerates 40+ named vars, two backends, and three new surfaces. v0.6.5

05 Web search grounding for ontology and facts rendering NEW

90

`WEB_SEARCH_ENABLED` (default `false`) turns on optional web grounding during ontology rendering and critique, firing only when node output requests it; `WEB_SEARCH_PROVIDER` selects the engine (`duckduckgo`); `WEB_SEARCH_TOP_K`, `WEB_SEARCH_TIMEOUT_SECONDS`, `WEB_SEARCH_MAX_SNIPPET_CHARS`, and `WEB_SEARCH_MAX_TOTAL_CHARS` control result count, timeout, and evidence volume. Per-stage toggles `WEB_SEARCH_ONTOLOGY_RENDER_ENABLED`, `WEB_SEARCH_ONTOLOGY_CRITIC_ENABLED`, `WEB_SEARCH_FACTS_RENDER_ENABLED`, and `WEB_SEARCH_FACTS_CRITIC_ENABLED` allow search-eligible retries per prompt type (first pass is always no-search). An LLM planner gated by `WEB_SEARCH_PLANNER_ENABLED`, `WEB_SEARCH_PLANNER_MAX_QUERIES`, `WEB_SEARCH_PLANNER_MIN_QUERY_CHARS`, and `WEB_SEARCH_PLANNER_MIN_CONFIDENCE` decides when to search; `WEB_SEARCH_REUSE_EVIDENCE_ACROSS_ATTEMPT` reuses node-scoped evidence across retries and `WEB_SEARCH_MIN_SNIPPET_CHARS` sets a minimum snippet length. `WEB_SEARCH_ALLOWED_DOMAINS` and `WEB_SEARCH_BLOCKED_DOMAINS` allow/block sources, and `WEB_SEARCH_REGION` / `WEB_SEARCH_SAFESEARCH` set DuckDuckGo region and safe-search mode.

Enable web grounding for ontology rendering and critic passes only, using DuckDuckGo, fetching the top 5 results, and blocking low-quality domains — paste into your shell or a .env file before running OntoCast.

```
$ export WEB_SEARCH_ENABLED=true
export WEB_SEARCH_PROVIDER=duckduckgo
export WEB_SEARCH_TOP_K=5
export WEB_SEARCH_ONTOLOGY_RENDER_ENABLED=true
export WEB_SEARCH_ONTOLOGY_CRITIC_ENABLED=true
export WEB_SEARCH_FACTS_RENDER_ENABLED=false
export WEB_SEARCH_BLOCKED_DOMAINS=spamsite.com,lowquality.net
export WEB_SEARCH_REGION=us-en
export WEB_SEARCH_SAFESEARCH=moderate
```

Tune the LLM planner to only fire searches when confidence is high and queries are substantive, limiting evidence injected per prompt to avoid context bloat.

```
$ export WEB_SEARCH_PLANNER_ENABLED=true
export WEB_SEARCH_PLANNER_MAX_QUERIES=3
export WEB_SEARCH_PLANNER_MIN_QUERY_CHARS=20
export WEB_SEARCH_PLANNER_MIN_CONFIDENCE=0.75
export WEB_SEARCH_MAX_SNIPPET_CHARS=400
export WEB_SEARCH_MAX_TOTAL_CHARS=2000
export WEB_SEARCH_MIN_SNIPPET_CHARS=80
export WEB_SEARCH_REUSE_EVIDENCE_ACROSS_ATTEMPT=true
```

— Enumerates all 18+ vars with defaults and stage-level behavior. product docs

06

Document conversion configuration and multi-format Docling ingestion

NEW

90

`CONVERTER_PROFILE` (`auto` | `fast` | `lean` | `ocr`) selects a conversion preset; `auto` picks per-PDF between `fast` (text layer) and `ocr` (image-only pages), and `lean` enables `CONVERTER_DO_FORMULA_ENRICHMENT` by default to decode display equations to LaTeX. `CONVERTER_LAYOUT_MODEL` (`heron` | `heron_101` | `egret_medium` | `egret_large` | `egret_xlarge` | `v2`), `CONVERTER_OCR_ENGINE` (`auto` | `easyocr` | `rapidocr` | `tesseract_cli` | `tesseract`), `CONVERTER_OCR_LANG` , `CONVERTER_PDF_BACKEND` (`docling_parse` | `pypdfium2`), and `CONVERTER_TABLE_MODE` (`accurate` | `fast`) select Docling's layout, OCR, PDF backend, and table modes. `CONVERTER_SUPPORTED_EXTENSIONS` (default includes `.pdf` , `.docx` , `.pptx` , `.xlsx` , `.html` , `.htm` , `.md` , `.csv` , `.adoc` , `.png` , and more) controls which suffixes Docling processes; ingestion was extended to Word,

Excel, HTML, Markdown, CSV, AsciiDoc, and images alongside existing PDF/PowerPoint support. `CONVERTER_DO_OCR`, `CONVERTER_DO_TABLE_STRUCTURE`, `CONVERTER_FORCE_FULL_PAGE_OCR`, `CONVERTER_TABLE_CELL_MATCHING`, and `CONVERTER_OCR_BITMAP_AREA_THRESHOLD` toggle/tune OCR and tables, while `CONVERTER_REPAIR_LIGATURE_GAPS`, `CONVERTER_REPAIR_NUMERIC_ARTIFACTS`, and `CONVERTER_FORCE_BACKEND_TEXT` repair publisher-PDF text artifacts.

Restrict Docling ingestion to only DOCX and Markdown files, skipping everything else with a warning.

```
$ CONVERTER_SUPPORTED_EXTENSIONS=.docx,.md ontocast process
```

— Names 16 conversion vars plus the newly supported file formats. v0.6.5

---

07 **Catalog management and server HTTP endpoints** NEW 88

The `ontocast serve` subcommand (with `--ontology-dir` and `--shapes-dir` flags) runs OntoCast as an HTTP server exposing: `GET /ontologies` to list each catalog ontology's current version, `POST /ontologies`, `PUT /ontologies/<url-encoded IRI>`, and `DELETE /ontologies/<url-encoded IRI>` to manage the running catalog, `POST /process_unit` to render a single content unit and inspect its ontology context in isolation, `GET /health` for liveness, `GET /info` for server capabilities and LLM cache counters, and `POST /flush` to delete data in a named partition. Per-request overrides are supported for `render_mode`, `ontology_context_mode`, `ontology_context_fixed_ontology_id`, `llm_graph_format`, and `max_visits`.

Upload a new ontology to a running server and immediately make it available for extraction without restarting.

```
$ curl -X POST http://127.0.0.1:8999/ontologies -F "file=@my-ontology.ttl"
```

Run OntoCast as a shared HTTP server backed by Fuseki, with per-document concurrency capped to avoid provider rate limits.

```
$ ontocast serve --ontology-dir ./my-ontologies --shapes-dir ./my-shapes
```

— Full method and path for every endpoint, runnable with curl. v0.6.5

---

08 **Fuseki triple-store connectivity and datasets** IMPROVED 87

OntoCast can persist graphs to an Apache Jena Fuseki server instead of in-memory storage: `FUSEKI_URI` sets the Fuseki HTTP root (e.g. `http://localhost:3030`), `FUSEKI_AUTH` supplies `user/password` or `user:password` credentials, and `FUSEKI_DATASET`, `FUSEKI_ONTOLOGIES_DATASET`, and `FUSEKI_SHAPES_DATASET` name the facts, ontologies, and SHACL-shapes datasets (auto-overridden by `ontocast serve` / `ontocast process` from tenant and project, shapes kept separate to avoid catalog-discovery collisions on `owl:Ontology` subjects). `ONTOCAST_ONTOLOGY_DIRECTORY` points at a directory of seed `*.ttl` files loaded once at startup. A later release made `FUSEKI_URI` alone sufficient to select Fuseki as the backend, making `FUSEKI_AUTH` optional.

Connect OntoCast to an authenticated local Fuseki instance and point it at your seed ontologies and a bounded LRU cache before running the server.

```
$ export FUSEKI_URI=http://localhost:3030
export FUSEKI_AUTH=admin:secret
export FUSEKI_DATASET=myproject_facts
export FUSEKI_ONTOLOGIES_DATASET=myproject_ontologies
export FUSEKI_SHAPES_DATASET=myproject_shapes
export ONTOCAST_ONTOLOGY_DIRECTORY=/opt/ontocast/ontologies
export ONTOCAST_CACHE_DIR=/var/cache/ontocast
export ONTOCAST_CACHE_MAX_BYTES=1GB
export ONTOCAST_CACHE_TTL_DAYS=30
export ONTOCAST_CACHE_PRUNE_EVERY=100
ontocast serve
```

— Names every var and the default-selection behavior change. v0.6.5

09

Ontology validation: reconciliation, critic passes, acceptance gates

NEW

85

`ONTOLOGY_RECONCILE_MINTED_TERMS` (`off` | `detect` | `rewrite`, default `detect`) catches and optionally replaces duplicate terms minted by the LLM renderer under partial vector-retrieval visibility. `ONTOLOGY_CRITIC_PASSES` (default 0) sets how many LLM review-and-patch passes run per ontology unit, with `ONTOLOGY_CRITIC_MAX_DELETE_SHARE` capping the share of a delta one pass may delete and `ONTOLOGY_CRITIC_MIN_DELETES` setting a floor of always-permitted deletions. `ONTOLOGY_ACCEPT_BLOCKING_FINDING_KINDS` (default `["foreign_delete", "foreign_namespace", "subclass_cycle", "role_confusion"]`) defines which deterministic findings block unit acceptance.

Enable two LLM critic passes per ontology unit while capping deletions at 10 % of the delta and always allowing at least 2 deletions, to catch structural errors without over-pruning.

📍 `ONTOLOGY_CRITIC_PASSES=2` `ONTOLOGY_CRITIC_MAX_DELETE_SHARE=0.1` `ONTOLOGY_CRITIC_MIN_DELETES=2` python `-m` ontocast …

Tighten acceptance rules to also block on `missing_label` findings in environments where every minted term must carry a label before merging.

📍 `ONTOLOGY_ACCEPT_BLOCKING_FINDING_KINDS` ='["foreign_delete", "foreign_namespace", "subclass_cycle", "role_confusion", "missing_label"]' python `-m` ontocast …

– Named vars with defaults and two runnable config examples. product docs

10 **Entity aggregation and disambiguation configuration** NEW

84

`AGG_EMBEDDING_MODEL` (default `sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`) sets the aggregation embedding model; `AGG_SIMILARITY_THRESHOLD` is the cosine threshold for the cross-graph aligner used by `POST /match/entities`, `match-graphs`, and the `ontocast_align_entities` agent tool, while `AGG_CANDIDATE_SIMILARITY_THRESHOLD` thresholds in-pipeline DBSCAN clustering. Merge guards include `AGG_LITERAL_CONFLICT_GUARD`, `AGG_INITIALS_DISTINCT_GUARD`, `AGG_NATURAL_KEY_MERGE`, and `AGG_TYPE_GUARD_UNTYPED` (`permissive` | `strict`); fuzzy lexical-alias merging uses `AGG_LEXICAL_LABEL_JACCARD`, `AGG_LEXICAL_SEQUENCE_RATIO`, and `AGG_LEXICAL_TOKEN_JACCARD`. `AGG_FUNCTIONAL_MIN_EMPIRICAL_SUPPORT` sets minimum observed-subject count for the functional-object guard, `AGG_SIBLING_GUARD_SCOPE` (`subject` | `predicate`) scopes the co-object sibling guard, and `AGG_UNIT_SCOPED_FACT_IRIS` suffixes minted fact IRIs with the originating unit index (`__u`) so independently minted IRIs become merge candidates.

– Names all 13 vars and two endpoints they feed; no runnable example given. product docs

11 **Document chunking and section classification configuration**

NEW

83

`CHUNK_MIN_SIZE` and `CHUNK_MAX_SIZE` set character bounds for chunk merging/splitting, and `CHUNK_MIN_UNIT_CHARS` hard-drops units below a character count. `CHUNK_EMBEDDING_MODEL` (default `sentence-transformers/paraphrase-multilingual-mpnet-base-v2`) selects the chunking/schema-detection model; `CHUNK_SEGMENTER` (`semantic` | `docling`) picks the built-in chunker or docling's `HybridChunker`. Section tagging is staged by `CHUNK_SECTION_CLASSIFIER` (`off` | `heading` | `heuristic` | `llm`, only `llm` calls the model), with `CHUNK_SECTION_TAG_MIN_CHARS`, `CHUNK_SECTION_TEXT_HEADINGS`, `CHUNK_SECTION_DENSITY` (`off` | `conservative` | `aggressive`), and

`CHUNK_SECTION_LLM_BATCH_SIZE` (`0` restores per-chunk calls). `CHUNK_SECTION_SCHEMA_DETECT` (`off` | `lexical` | `headings` | `auto`) infers document type, tuned by `CHUNK_SECTION_SCHEMA_DETECT_MIN_SCORE`, `_MIN_MARGIN`, and `_CONTENT_MIN_MARGIN`. `CHUNK_SECTION_FILTER_ON_EMPTY` (`warn` | `error`) can fail batch runs with HTTP 422. `CHUNK_BIBLIOGRAPHY_MODE` (`skip` | `citations_only` | `domain_facts`) routes reference-list chunks, with `CHUNK_CITATION_VOCABULARY` overriding schema.org CURIEs for bibo/FaBiO/DCMI catalogs. `CHUNK_NON_CONTENT_MODE` (`extract` | `skip`) handles front/back matter, and `CHUNK_MAX_MEASUREMENTS_PER_UNIT` recursively splits measurement-dense units at sentence midpoints.

— Enumerates 16 vars with exact option sets; no runnable example given. product docs

12 **Config file layering and auditing** NEW 83

`ontocast --env-file FILE` (repeatable) loads settings files in order, with later files and shell-exported variables taking final precedence. The `ontocast config check FILE...` subcommand classifies variables in a settings file as unknown, invalid, redundant, or pinned, exiting 1 on unknown or invalid entries. Config.in_memory() builds a configuration requiring no external services for library use, and `ontocast.config.env_names` exposes the environment-variable name of every settings field.

Layer a base config and a per-environment override so team secrets in the environment always win, without editing the shared file.

```
$ ontocast --env-file .env.base --env-file .env.prod run
```

Audit a settings file before deploying to catch renamed or removed variables that would silently do nothing.

```
$ ontocast config check .env.prod
```

Layer credentials and project settings separately, then override the model for a single run without touching either file.

```
$ LLM_MODEL_NAME=gpt-5.6-terra ontocast \
  --env-file ~/.config/ontocast/openai.env \
  --env-file project.env \
  process --input-path ./papers --output-dir ./out
```

Audit a project env file before deploying to catch renamed, invalid, or redundant variables without exposing their values.

```
$ ontocast config check project.env
```

— Names exact flags/subcommand/method with runnable audit examples. v0.6.6 v0.6.5

13 **Disk cache for LLM responses and tool outputs** NEW

81

`ONTOCAST_CACHE_DIR` sets the cache directory for LLM responses and tool outputs; `ONTOCAST_CACHE_MAX_BYTES` caps total size with LRU eviction (accepts byte counts or strings like `'1GB'`, `0` disables pruning); `ONTOCAST_CACHE_TTL_DAYS` evicts entries unused for N days before the size ceiling applies; `ONTOCAST_CACHE_PRUNE_EVERY` controls how many writes trigger a size re-check. The `ontocast cache` CLI command adds `stats`, `prune`, and `clear` subcommands to inspect, trim, or empty the cache.

Connect OntoCast to an authenticated local Fuseki instance and point it at your seed ontologies and a bounded LRU cache before running the server.

```
$ export FUSEKI_URI=http://localhost:3030
export FUSEKI_AUTH=admin:secret
export FUSEKI_DATASET=myproject_facts
export FUSEKI_ONTOLOGIES_DATASET=myproject_ontologies
export FUSEKI_SHAPES_DATASET=myproject_shapes
export ONTOCAST_ONTOLOGY_DIRECTORY=/opt/ontocast/ontologies
export ONTOCAST_CACHE_DIR=/var/cache/ontocast
export ONTOCAST_CACHE_MAX_BYTES=1GB
export ONTOCAST_CACHE_TTL_DAYS=30
export ONTOCAST_CACHE_PRUNE_EVERY=100
ontocast serve
```

Inspect and prune the on-disk cache to reclaim space after bulk extraction runs.

```
$ ontocast cache stats
ontocast cache prune
```

— All cache env vars plus the CLI subcommands to act on them. product docs

14 **CLI command suite and modular install extras** NEW

76

OntoCast exposes CLI commands for starting the server, extracting from a file or directory, printing a document outline, batch-converting PDFs to Markdown/JSON, disambiguating entities in a Turtle graph, and regenerating workflow diagrams via `plot-graph`, which supports `--format` as `svg`, `png`, or `pdf` (default `svg,png`).

Installation is modular via pip extras such as `server`, `openai`, `doc-processing`, and `lancedb`, down to an embedding-only footprint (extraction pipeline, RDF stack, in-memory triple/vector stores, and ontology tooling only) for library integration.

Install OntoCast with full document-processing, server, and OpenAI support to handle PDF, Office, HTML, Markdown, and image inputs.

```
$ pip install "ontocast[server,openai,doc-processing]"
```

Export a knowledge graph in all three formats at once for inclusion in reports and presentations.

```
$ ontocast plot-graph --format svg,png,pdf
```

— Names the install extras and the plot-graph flag with values. v0.6.5

---

15 **Partial-document evaluation via --head-chunks** NEW

73

The `--head-chunks` flag on `ontocast process` stops processing after a specified number of content units/document parts, enabling fast low-cost evaluation runs and cost-vs-scale measurement against the LLM response cache without paying for LLM tokens twice.

Measure how CPU-side processing cost grows with document length without paying for LLM tokens a second time.

```
$ ontocast process --input-path doc.pdf --head-chunks 5 --output-dir ./out
ontocast process --input-path doc.pdf --head-chunks 20 --output-dir ./out
```

Quickly evaluate OntoCast on a new document by processing only the first three parts to judge output quality before committing to a full run.

```
$ ontocast process --input-path ./paper.pdf --head-chunks 3 --output-dir ./out
```

— Exact flag with runnable before/after commands. product docs

---

16 **Shared local embedding model across chunking, retrieval, and aggregation**

IMPROVED

72

`CHUNK_EMBEDDING_MODEL`, `EMBEDDING_MODEL_NAME`, and `AGG_EMBEDDING_MODEL` are shared process-wide when set to the same model name, halving resident local-model memory on constrained hosts.

Consolidate all three local embedding uses to a single model to halve resident memory when running on a constrained host.

```
$ export CHUNK_EMBEDDING_MODEL=sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2
export EMBEDDING_MODEL_NAME=sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2
export AGG_EMBEDDING_MODEL=sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2
ontocast process --input-path doc.pdf --output-dir ./out
```

— Names all three vars and the concrete memory-saving mechanism. product docs

---

17 **Default model and provider preset changes** BREAKING

70

The default model changed to `gpt-5.6-luna` (was `gpt-5.4`); cached LLM entries for `gpt-5.4` are not reused. Claude presets now use provider API IDs, and the legacy `claude-4.x-*-latest` aliases are removed. Gemini 2.x and 1.5 presets are removed, though other Gemini names still pass through with a warning.

— Names old and new defaults and removed aliases, no migration command. v0.6.6

---

18 **Per-run seed ontology directory overrides** NEW

68

`--ontology-dir` on `ontocast process` specifies (or clears with `''`) the seed ontology directory for a single run, independent of `ONTOCAST_ONTOLOGY_DIRECTORY`, and `--ontology-output-dir` directs ontology output to a separate directory.

— Names two exact flags and their relation to an existing env var. product docs

---

19 **Run manifest and named output files** NEW

67

Processing runs write `out/paper.run.json` recording actual settings used, LLM call counts, cache hits, and token budget, alongside `paper.facts.ttl`, `paper.<ontology>.ontology.ttl`, and `paper.facts.validation.json` as named output files.

— Names exact output filenames but no example of reading them. product docs

---

20 **Conversion profile presets narrowed, suthing extra dropped**

63

BREAKING

`CONVERTER_PROFILE` values `born_digital` and `default` are removed; use `auto`, `fast`, or `ocr` (plus the newer `lean`) instead. The `suthing` package is removed from the `server` install extra.

— Names the removed values and dependency but no migration script. v0.6.5

---

21 **Graph serialization format selection** NEW 62

`LLM_GRAPH_FORMAT` selects the graph syntax the model writes during rendering — `jsonld` (default) or `turtle` — with `turtle` spending fewer characters per triple, reducing prompt size.

Extract facts only against a curated catalog pinned by ID, failing fast if the ontology context is missing, with prompt-friendly Turtle chapter format.

```ini
RENDER_MODE=facts
ONTOLOGY_CONTEXT_MODE=fixed_single_ontology
ONTOLOGY_CONTEXT_FIXED_ONTOLOGY_ID=https://example.org/ontologies
/my-catalog
ONTOLOGY_CHAPTER_FORMAT=term_sheet
ONTOLOGY_CONTEXT_REQUIRED=true
LLM_GRAPH_FORMAT=jsonld
```

— Names the var and both values with a stated trade-off. product docs

└──▷ **BREAKING ON UPGRADE**

! Default model changed to `gpt-5.6-luna` (was `gpt-5.4`); cached LLM entries for `gpt-5.4` are not reused.

! Claude presets now use provider API IDs; legacy `claude-4.x-*-latest` aliases are removed.

! Gemini 2.x and 1.5 presets removed; other Gemini names still pass through with a warning.

! `CONVERTER_PROFILE` values `born_digital` and `default` are removed; use `auto`, `fast`, or `ocr` instead.

! `suthing` is removed from the `server` extra.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261004 NOTES ↗

RANK

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

> LangSmith's window centers on infrastructure and access-control hardening — server-sent events for thread queries, expanded OAuth credential sharing for agents, IAM-based ECR auth in Sandboxes, and a breaking change tightening dataset export permissions — alongside incremental UI and Playground additions.

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

01 Provider tagging for Gateway model configs NEW

83

OpenAI-compatible model configurations can set an `ls_provider` field to populate trace metadata and match custom models to provider-specific pricing through the LLM Gateway.

Tag a custom OpenAI-compatible model with its provider so LLM Gateway traces show correct metadata and pricing.

```json
{
  "model": "my-org/llama-3-70b",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "meta"
}
```

— Named config field with a runnable example. snapshot-20261004

02 Configurable license expiration warning window NEW

80

Self-hosted operators can set the `LICENSE_EXPIRATION_WARNING_DAYS` environment variable to control how many days before expiration the warning banner appears, replacing the previous fixed default of 7 days.

Give self-hosted teams earlier notice of license expiration by showing the banner 30 days out instead of the default 7.

```env
LICENSE_EXPIRATION_WARNING_DAYS=30
```

— Exact env var with a runnable example value. snapshot-20261004

03 **Playground model additions and Bedrock request tuning**

IMPROVED 73

Adds `Claude Fable 5.1` to the Anthropic, Bedrock, and Vertex AI model selectors in the playground. Amazon Bedrock models in the Playground now expose request timeout and max retries options, preventing long-running agent calls from being cut off at the AWS client default of 60 seconds.

— Named model and limit but UI-only options, no config key. snapshot-20261004

04 **OAuth authorization session and credential sharing** IMPROVED

73

OAuth authorization sessions add `owner_type` and `owner_id` parameters so a managed credential can be authorized for either a user or an agent, and now also accept workspace owners so one managed OAuth credential can be shared across a workspace. Agent OAuth connections show a provider-branded result page after authorization, and Gateway requests authenticated with a session or OAuth bearer token retain authentication and workspace context across internal routing, including model fallbacks.

— Named parameters but no example request given. snapshot-20261004

05 **Thread-level dataset and annotation queue actions** NEW

70

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

— UI entry points named but no exact navigation path. snapshot-20261004

06 **IAM role-based ECR auth in Sandboxes** NEW 62

Sandboxes now support AWS IAM role assumption on demand for commercial private ECR registries, enabling repository and tag search without relying on expiring saved authorization tokens.

— Mechanism described but no concrete flag or command given. snapshot-20261004

07 **LLM Gateway fallback on missing response headers** IMPROVED

60

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

— Mechanism explained but timeout config not named. snapshot-20261004

THINNER COVERAGE BELOW

08 **MCP connector support for Client ID Metadata Documents** NEW

55

The LangSmith MCP connector now supports Client ID Metadata Documents for authentication, retaining dynamic client registration as a compatibility fallback.

— Named mechanism but no setup steps given. snapshot-20261004

09 **Dedicated service for trajectory batch reads** IMPROVED

55

Trajectory batch reads now run on a dedicated in-cluster gRPC service instead of the smith-go monolith, so trajectory evaluation load no longer competes with UI and API traffic.

— Infra change described but nothing user-invokable. snapshot-20261004

10 **IssueBench cost and trajectory regression benchmark** NEW

50

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

— Named benchmark but no usage instructions. snapshot-20261004

11 **Streaming thread query results via SSE** IMPROVED 45

Thread query clients now support server-sent events, streaming results progressively while preserving cursor pagination.

— No endpoint or flag named for the SSE mechanism. snapshot-20261004

12 **Workspace switcher accepts workspace IDs** IMPROVED 30

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

— Minimal description with no example or location. snapshot-20261004

└──▷ **BREAKING ON UPGRADE**

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

WAS THIS USEFUL? 👍 👎

eunomia-bpf AgentSight

Feedback ⓘ v1.0.32 NOTES ↗

RANK

lightweight system-level observability for AI Agents

> AgentSight added support for monitoring CodeBuddy CLI agent sessions, extending its live attachment and transcript capture to a new agent type.

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

01 CodeBuddy CLI session monitoring

NEW INTEGRATION NEW

78

AgentSight now supports `CodeBuddy CLI` (`@tencent-ai/codebuddy-code`) as a monitored agent, reading session files from `~/.codebuddy/projects/<project>/<session-id>.jsonl`. The config home directory can be overridden via the `CODEBUDDY_CONFIG_DIR` environment variable, and `agentsight top` plus the existing transcript readers consume CodeBuddy sessions the same way they already read Claude, Codex, and Cursor sessions.

— Names exact paths, env var, and command but no example shown. product docs

v1.0.32 NOTES ↗ LATEST STABLE

Traces resolved file access paths in the eBPF process probe, providing visibility into actual agent file access after path resolution.

└──▷ **GET THIS VERSION**

```
$ git clone --branch v1.0.32 https://github.com/eunomia-bpf/agentsight.git
# already have the repo? check out this version:
$ git checkout v1.0.32
```

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261004 NOTES ↗ RANK

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

> Braintrust shipped project-group-aware project creation, new AgentScope tracing support, hardened Windows file permissions for tracing, and new `bt trace` plugin-management commands, alongside a credential-loss issue when downgrading the CLI on Windows.

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

01 Project group placement for new projects NEW 75

`bt functions push` now recognizes `projectGroupName` / `project_group_name` so that newly created projects are automatically placed in the correct project group; the same can be set directly via client.projects.create(name=..., project_group_name=...) when creating a project through the SDK.

Create a new project and place it inside an existing project group in one call, useful for keeping agent-specific projects organized.

python

```python
from braintrust import Braintrust

client = Braintrust()
client.projects.create(name='my-coding-agent',
project_group_name='Platform Team')
```

— Names exact config field and API call but no deeper mechanism explained

snapshot-20261004

02 bt trace doctor and plugin update commands NEW 75

`bt trace doctor` checks whether tracing plugins such as the Claude Code tracing plugin are out of date, and `bt trace update claude` updates it; relevant to check after upgrading the `bt` CLI to v0.22.2.

Check that the Claude Code tracing plugin is up to date after upgrading bt to v0.22.2, then update it if flagged as out of date.

```
$ bt trace doctor
bt trace update claude
```

— Exact runnable commands given but no further mechanism detail snapshot-20261004

THINNER COVERAGE BELOW

03 Owner-only permissions for Windows tracing files IMPROVED

50

Tracing settings files, the tracing data directory, and the background tracing process's named pipe on Windows are now accessible only to the current user and SYSTEM, matching the owner-only permission model already used on macOS and Linux.

— Explains before/after scope but needs no reader action `snapshot-20261004`

04 Tracing support for AgentScope TeamPipeline streaming NEW

50

Adds tracing of AgentScope TeamPipeline.reply_stream() calls in the Python SDK, requiring AgentScope v2.0.9 or later.

— Names method and version requirement but no usage example `snapshot-20261004`

05 Credential loss when downgrading bt CLI on Windows BREAKING

50

After downgrading the `bt` CLI from v0.22.2 to an earlier version on Windows, saved credentials are lost and the user must log in again.

— States exact affected versions and effect, no workaround given `snapshot-20261004`

└──▷ **BREAKING ON UPGRADE**

! After downgrading the `bt` CLI from v0.22.2 to an earlier version on Windows, saved credentials are lost and you must log in again.

WAS THIS USEFUL?  

AI MODELS

◆ Open-Weight Models

DeepSeek V3.1 Terminus

Feedback ⓘ

price cut · Oct 3, 2026 MODEL ↗

DeepSeek V3.1 Terminus is a chat model from DeepSeek.

DeepSeek V3.1 Terminus pricing drops 10% on OpenRouter.

Price cut on OpenRouter (deepseek/deepseek-v3.1-terminus): $0.30 → $0.27 in, $1.00 → $1.00 out per million tokens (-10%), first seen 3 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.1

Feedback ⓘ price increase · Oct 3, 2026 MODEL ↗

GLM 5.1 is a chat model from Z.ai.

Price increase on OpenRouter (z-ai/glm-5.1): $0.9646 → $1.40 in, $3.0316 → $4.40 out per million tokens (+45%), first seen 3 Oct 2026.

WAS THIS USEFUL? 👍 👎

Google Gemma 4 26B A4B

Feedback ⓘ

price cut · Oct 3, 2026 MODEL ↗

Gemma 4 26B A4B is a multimodal model from Google.

Price cut on OpenRouter (google/gemma-4-26b-a4b-it): $0.09 → $0.0675 in, $0.30 → $0.225 out per million tokens (-25%), first seen 3 Oct 2026.

WAS THIS USEFUL? 👍 👎

Moonshot AI Kimi K2.6 Feedback ⓘ

price increase · Oct 3, 2026 MODEL ↗

Kimi K2.6 is a multimodal model from Moonshot AI.

Price increase on OpenRouter (moonshotai/kimi-k2.6): $0.4341 → $0.95 in, $1.828 → $4.00 out per million tokens (+119%), first seen 3 Oct 2026.

WAS THIS USEFUL? 👍 👎

NVIDIA Nemotron 3.5 Lightning Feedback ⓘ

context change · Oct 3, 2026 MODEL ↗ STABLE

Nemotron 3.5 Lightning is a chat model from NVIDIA.

Context window shrank from 1M to 262K tokens on OpenRouter (nvidia/nemotron-3.5-lightning), first seen 3 Oct 2026.

▾ 1 more release in this issue · 2026-10-03

> price cut · Oct 3, 2026 MODEL ↗ STABLE
>
> Price cut on OpenRouter (nvidia/nemotron-3.5-lightning): $0.08 → $0.0595 in, $0.20 → $0.17 out per million tokens (-26%), first seen 3 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 Next 80B A3B Thinking

Feedback ⓘ

retirement scheduled · Oct 3, 2026 MODEL ↗

Qwen3 Next 80B A3B Thinking is a reasoning model from Alibaba.

OpenRouter lists a retirement date of 9 Oct 2026 for qwen/qwen3-next-80b-a3b-thinking, first seen 3 Oct 2026.

WAS THIS USEFUL? 👍 👎

◆ Frontier Models

Anthropic

Feedback ⓘ changelog · Oct 4, 2026 CHANGELOG ↗ RANK

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

> Anthropic shipped Claude Sonnet 5.5 on the API, alongside several breaking changes to thinking-mode controls and tool-use behavior that developers migrating from Claude Sonnet 5 need to handle, plus the deprecation of Claude Sonnet 4.5.

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

01 Thinking-mode disable syntax changes

BREAKING 90

Turning off up-front thinking at or below a given effort level in Claude Sonnet 5.5 now requires sending `thinking: {"type": "between_tools"}` instead of the previous `thinking: "disabled"` value.

Disable up-front thinking in a Claude Sonnet 5.5 request — required syntax change from the old `'disabled'` value.

```
$ 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-sonnet-5-5",
    "max_tokens": 1024,
    "thinking": {"type": "between_tools"},
    "messages": [{"role": "user", "content": "Summarise this
report."}]
  }'
```

— Exact flag syntax plus a runnable curl example changelog-20261004-448179c9

02 Tool-use behavior changes in Sonnet 5.5

BREAKING 75

Forced tool use on Claude Sonnet 5.5 now returns a 400 error; the advisor tool rejects Claude Opus 4.8, Claude Opus 4.7, and Claude Sonnet 5 as valid advisors; and the earlier computer use tool is no longer accepted on the Claude API or Google Cloud when using Claude Sonnet 5.5.

— Three named behavior changes but no commands to apply fixes changelog-20261004-448179c9

03 Thinking blocks scoped to account and conversation

BREAKING

60

Thinking blocks produced by Claude Sonnet 5.5 are tied to the model and the conversation: blocks sent from an unlinked account are silently dropped before the model sees them, without causing the request to fail.

— Mechanism explained but no command to act on `changelog-20261004-448179c9`

THINNER COVERAGE BELOW

04 Claude Sonnet 4.5 deprecated DEPRECATED

55

Claude Sonnet 4.5 is deprecated and scheduled for retirement on the Claude API on November 30, 2026; callers should migrate to `claude-sonnet-5-5`.

— Gives exact retirement date and migration target model id `changelog-20261004-448179c9`

05 Claude Sonnet 5.5 added to the API NEW

40

Adds Claude Sonnet 5.5 (`claude-sonnet-5-5`) to the Claude API, with updated context window, output limits, and pricing compared to prior Sonnet models.

— Names the model id but no concrete context/pricing numbers `changelog-20261004-448179c9`

└──▷ **BREAKING ON UPGRADE**

! Code targeting Claude Sonnet 5 can break on Claude Sonnet 5.5 in five documented ways: (1) `thinking: "disabled"` must be replaced with `thinking: {"type": "between_tools"}` to turn off up-front thinking; (2) forced tool use now returns a 400 error; (3) thinking blocks are tied to the model and the conversation; (4) the earlier computer use tool is not accepted on the Claude API or Google Cloud; (5) the advisor tool rejects Claude Opus 4.8, Claude Opus 4.7, and Claude Sonnet 5 as advisors.

! Claude Sonnet 4.5 is deprecated and scheduled for retirement on the Claude API on November 30, 2026; callers should migrate to `claude-sonnet-5-5`.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 4, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue