

Preflight Adds a Sanitizing Boundary to the Multi-Model Agent Stack

Coding Agents Alpha Tracker

2026-09-21

Preflight Adds a Sanitizing Boundary to the Multi-Model Agent Stack

By Coding Agents Alpha Tracker • September 21, 2026

A new gateway pattern puts redaction, local attachment inspection, and model routing between coding agents and their providers; Jev's latest practitioner guidance clarifies the narrow role of fast typed decisions.

TOP SIGNAL

Geoffrey Huntley's **preflight** makes the agent gateway—not a prompt—the security boundary. His chain is `client → preflight → underclass → model`: Underclass collapses 20 ChatGPT/Copilot subscriptions into one endpoint, while Preflight listens on `:8081` and starts with `nix run github:ghuntley/preflight -- serve`. [1, 2] **It defaults to redaction rather than aborting—ghp_...** becomes `[REDACTED:github-pat]`—while screenshots, PDFs, barcodes, and OCR are decoded locally; unsanitized attachments do not go upstream, and logs/HTTP 409s expose only finding and rule IDs. [3, 4]

TRY THIS

- **Put a security hop in front of the harness.** Point the agent from `:8080` to `:8081`, run `nix run github:ghuntley/preflight -- serve`, and keep the `client → preflight → underclass → model` path. Make repeated multimodal checks reusable by caching the verdict for the same attachment plus inspection profile content-addressably on disk for a one-week TTL; vendor and pin Gitleaks, update it through a CI-checked PR, and keep new rules disabled until explicitly allowed in `defaults.toml`. [1, 5, 6]

- **Use Jev as a typed fast lane, not an agent.** Pass structured state and a schema into it, use its confidence to set an automation threshold, and send below-threshold cases to review; keep codebase traversal, tool outputs, and history compaction in the reasoning model. Theo describes Jev as a function-like classifier with a 32k-token context and explicitly rejects using it to judge several LLM outputs or compact agent history; Salvatore Sanfilippo gives the confidence-to-human-review pattern. [7, 8]
- **Provision keys out of band on remote agent machines.** Simon Willison’s `llm-keys-ui` workflow is: have Codex run `uvx --with llm-keys-ui llm keys-ui --all`, open the returned local-network or Tailscale URL to save the key, then let a later shell command retrieve it with `llm keys get anthropic` instead of pasting the secret into the agent session. [9]
- **Use a project as the orchestration folder.** In Riley Brown’s Claude Project pattern, name the project, give it a goal, let the main chat create task-specific threads, and have those threads—not the main agent—use tools. Assign models per thread, store their artifacts in the project library, and keep recurring routines inside the project so new threads inherit the project context without re-explaining the work. [10]

WHAT SHIPPED

- **llm-keys-ui 0.1:** Simon Willison released the web UI for setting LLM keys, motivated by controlling Codex Remote sessions from a phone without pasting API keys into the agent or ChatGPT app. [9]
- **Agent gateway pair — underclass and preflight:** Underclass is Huntley’s one-endpoint consolidation of 20 ChatGPT/Copilot subscriptions; Preflight adds the redacting, attachment-aware checkpoint in front of it. [2, 1]
- **OpenClaw FaceTime support:** Omar Shahine says the experimental plugin supports audio, lets users call their claw and the claw call them, supports GPT-live, and should arrive in the next release; Peter Steinberger separately announced that a claw can now FaceTime, so treat the rollout status as qualified. [11, 12]
- **Jev early access:** Theo’s walkthrough positions Typesafe AI’s model as structured state in, typed probabilistic decisions out, available through OpenRouter and Vercel AI Gateway. He cites 70–500 ms classification and roughly \$0.04 per million tokens versus \$10 for Fable, but stresses the 32k-token context and Jev’s unsuitability for codebase traversal, LLM judging, or context compaction. [7]
- **Kody’s shared capability layer:** Kent C. Dodds is positioning Kody as a common place for multiple assistants to run integrations, secrets, automations, jobs, webhooks, and software packages; he describes the

package feature as a personal software registry plus execution environment. In a separate firsthand example, his Grok bot used Kody’s calendar and email connections to manage a trip and draft a reply. [13, 14, 15]

GO DEEPER

- **Video — Theo’s Jev is incredible:** Watch the model-boundary segment first: Jev is a function-like classifier for structured state, not a code-writing model. The later section is the useful anti-pattern review—why judging multiple LLM outputs or compacting agent history throws away the context and tool results those tasks require. [7]



Jev is incredible (5:56)



Jev is incredible (22:25)

- **Repo — preflight + underclass:** Study the proxy boundary, default-redaction behavior, local multimodal inspection, content-addressed verdict cache, and CI-controlled Gitleaks rule updates before copying the pattern. [1, 4, 5, 6]
- **Repo — llm-keys-ui 0.1:** A small, immediately reusable pattern for keeping API-key entry in a separate web surface while remote coding agents retrieve keys only at command execution time. [9]

Editorial take: The practical frontier is boundary design: sanitize model calls, keep narrow decisions typed, and reserve human time to inspect generated work instead of optimizing only for “press enter.” [1, 7, 16]

Sources

1. X post by @GeoffreyHuntley
2. X post by @GeoffreyHuntley
3. X post by @GeoffreyHuntley
4. X post by @GeoffreyHuntley
5. X post by @GeoffreyHuntley
6. X post by @GeoffreyHuntley
7. Jev is incredible

8. Chi se ne frega di Jev?
9. llm-keys-ui 0.1
10. X post by @rileybrown
11. X post by @OmarShahine
12. X post by @steipete
13. X post by @kentcdodds
14. X post by @kentcdodds
15. X post by @kentcdodds
16. X post by @v0xium