

Orlo

Models change. Your standard doesn't.

Orlo is an evaluation and control layer for production AI. You set a standard once: examples, rules, and the documents answers can draw on. Orlo tests any model against it, releases the winner as a frozen version, checks every response, and decides what each agent step may do.

The problem

Providers update and retire models on their own schedule, and public benchmarks don't measure your task. A replacement can quietly change an output format or an accuracy level. When an AI system makes a decision, most teams can't say which model, prompt, rules, and sources produced it.

What Orlo does

- **Tests models on your data.** Candidates from different providers, including models you host, run on the same examples with the same scoring, and a judge model that is never a candidate scores against a rubric you write. Results carry 95% confidence intervals. When two models are within each other's margin of error, Orlo says they are too close to call.
- **Freezes what goes live.** A release fixes the model, prompt, rules, and whether it answers from your documents. Every answer is logged against its release, so "what was running?" has an exact answer.
- **Checks every response.** Rules on schema, required and forbidden fields, patterns, and length run before a response is returned. A response that fails can be rejected or retried on a fallback model.
- **Answers from your documents and shows which.** Orlo searches a task's documents by keyword, and by meaning when an embedding service is set up, gives the model the passages it finds, and records which ones each answer used. When nothing matches, the model is told so.
- **Decides what agent steps may do.** Apart from three built-in control tools, agent tools are denied unless a rule allows them. Sessions have limits on steps, tool calls, and writes. A risky step waits for a person, and each approval records who decided. Orlo decides and records; your agent runtime enforces.
- **Keeps the standard as things change.** Reviewer corrections become new examples. When a model is retired or a better one ships, you run the same comparison on the replacement.

What Orlo isn't

- **A model host.** It calls hosted providers, or endpoints you run with vLLM or llama.cpp.
- **An agent framework.** It never runs your tools or holds credentials for your systems.
- **A compliance system.** It records evidence that risk and compliance teams can use. It doesn't replace their registers, assessments, or sign-off.
- **A watcher.** It doesn't track provider retirement dates. You decide when to re-test.

Who it's for

- **Applied-ML and AI infrastructure teams** that own model choice across several tasks.
- **Domain teams** in fraud, support, claims, and document review with a task AI could take on and no one to evaluate it.
- **Organizations that must run on their own infrastructure** and show what ran.

Try it

Sandbox. A managed sandbox is open to any organization. Bring one task and its labeled examples, compare two models, and release the winner. Start at docs.useorlo.com.

Self-hosted. Orlo runs with Docker Compose on your own infrastructure, with Postgres and Redis. The deployment guide lists the secrets it needs.

Under the hood

Evaluation. Bootstrap intervals from 1,000 resamples, early drop of models that are clearly behind, and hard token and runtime budgets. **Isolation.** Per-organization provider keys encrypted with AES-256-GCM, and row-level security on every organization table. **Runtimes.** OpenAI, Anthropic, Cloudflare Workers AI, Hugging Face, vLLM, and llama.cpp. **Interfaces.** A REST API, a web console, and embeddable components.