

(Invariant Extraction + Monitoring)

It has three stages working in sequence: **extract** machine-checkable invariants from a natural-language spec using an LLM, **instrument** a running system to log them, then **alert** when any invariant is violated in production — turning your prose spec into a live runtime monitor.

Tech stack

Extraction layer — Python script calling OpenAI, Anthropic, and Gemini APIs in parallel. Output normalized to a shared Pydantic schema. Store in a SQLite or Postgres table with versioning so you can diff invariant sets across spec versions.

Evaluation + logging — Each check emits a structured event to a message queue (Kafka or SQS). A consumer writes to your observability backend (Datadog, Grafana Loki, or even a plain Postgres table).

Dashboard + alerting — Grafana dashboard with panels for violation rate per invariant, coverage heatmap (which invariants have zero events in the last 24h), and a severity matrix. Alerts trigger a Lambda/Cloud Function that calls the LLM to produce the natural-language explanation before routing to Slack/PagerDuty

[Live link](#)

[github link](#)

