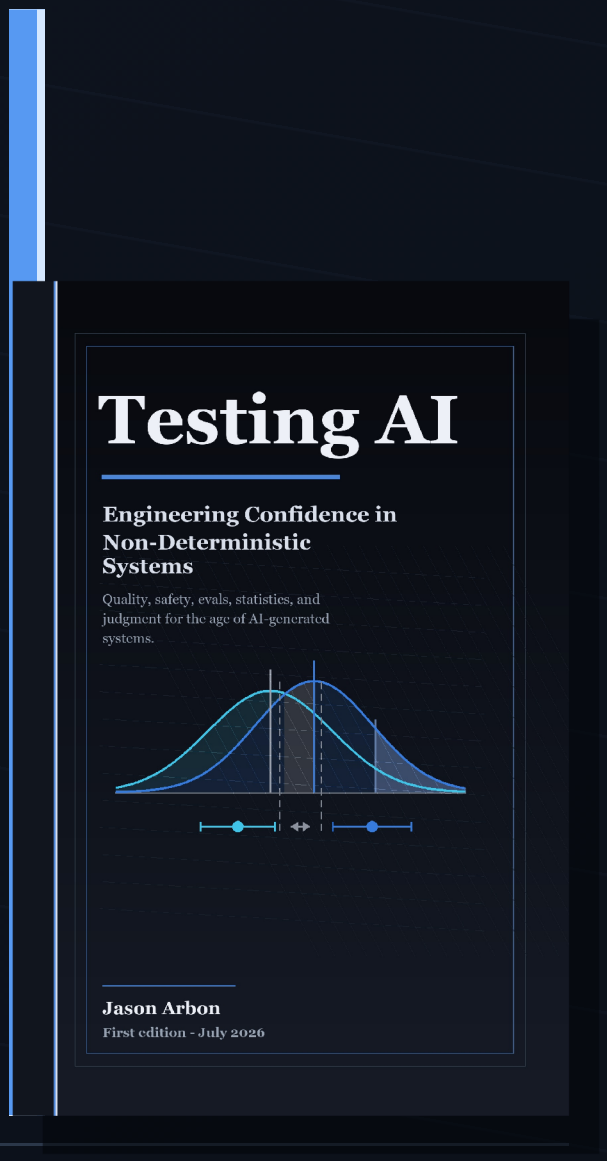
OPERATING AI

Quality has to survive the real world.

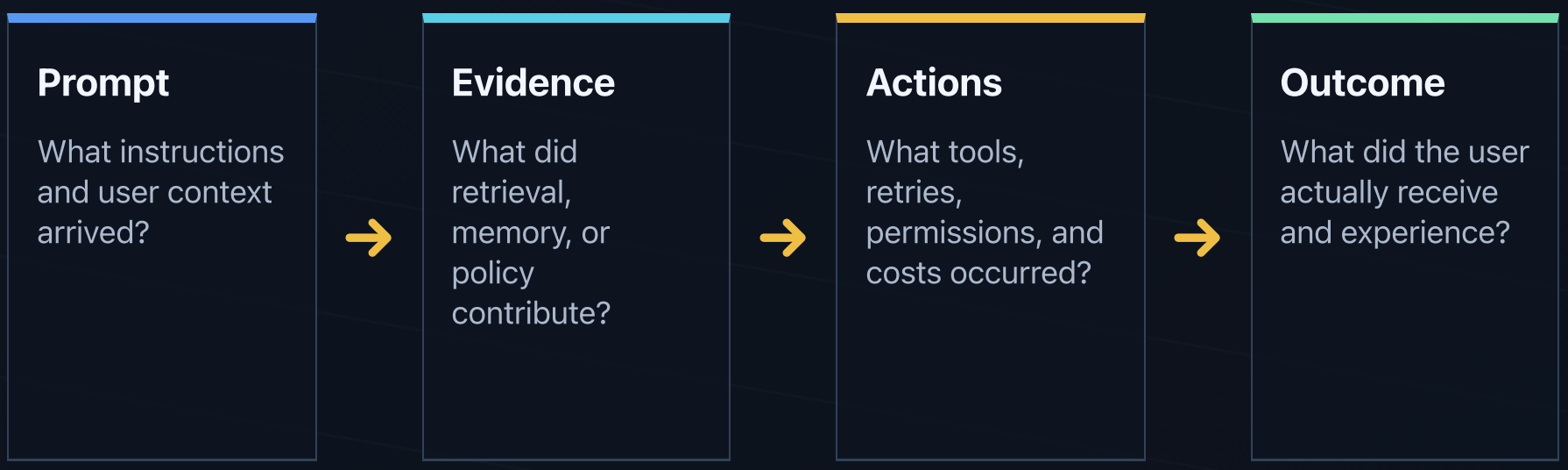
Production traces, retrieval, model routes, configuration, rollout controls, and cost all shape the answer users receive.

Swipe for the operational evidence behind trustworthy AI.



OBSERVABILITY

You cannot fix a final answer if you cannot see the path.



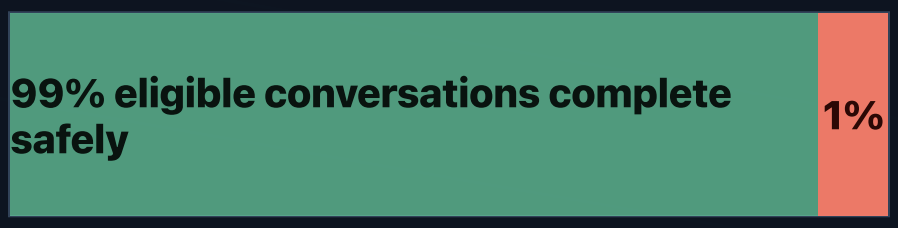
> A trace is not a dashboard decoration. It is the evidence trail that turns a surprising result into a reproducible fix.

Connect failing evals to traces, and promote meaningful production failures back into the eval suite.

SERVICE OBJECTIVES AND INCIDENT RESPONSE

Do not average a catastrophic failure into a green dashboard.

One SLO, one error budget



Track several user-visible objectives: quality, safety, latency, tool reliability, and cost. Severe privacy or irreversible-action failures need their own near-zero budget.

Detect weak signals

Contain the risky route

Preserve evidence

Assess user impact

Recover, notify, learn

Every meaningful incident should become a versioned regression case, monitor, and recovery drill.

RAG QUALITY

A grounded answer needs both the right evidence and faithful use of it.

Retrieval quality

- Needed source and chunk found
- Current, allowed, non-duplicated context
- Authority, tenant, and freshness checks
- Missing evidence is visible

Generation quality

- Claims supported by the context
- Citations point to actual support
- Conflicts and uncertainty handled honestly
- Abstains when evidence is missing

> A fluent answer can be perfectly grounded in a stale policy. Separate retriever failures from generator failures.

Score document freshness, retrieval hit rate, context precision, answer faithfulness, and citation support separately.

EVERY CHANGE IS A RELEASE

The model is only one part of the versioned bundle.

Prompt

Instructions, examples, and assembly logic.

Policy

Rules, allowed behavior, and refusal language.

Retrieval

Index snapshot, filters, reranker, and sources.

Tools

Schema, permissions, error shape, and contracts.

Evaluator

Judge prompt, rubric, labels, and scoring code.

Model route

Provider, model version, fallback, and region.

> If any of these change behavior, users experienced a product release. Version and test them together.

The most dangerous regression may come from the file someone dismissed as “just data.”

SHADOW, CANARY, PROMOTE

New behavior should earn traffic gradually.

Shadow

Fork real requests to the next version. Current production still serves the user.

Canary

Expose a small, intentionally chosen user slice. Watch it separately.

Promote

Expand only after quality, safety, cost, latency, and escalation evidence remains stable.

Write rollback triggers before rollout. Shadow gives realistic evidence; canary limits harm; neither replaces a planned measurement strategy.

A canary is a risk-control mechanism. A randomized experiment is a different tool for causal measurement.

FROM THE FIELD

Rollback is product code. Test it like it can become the outage.

New instances fail

A rollout used more production memory than expected. Instances booted, got traffic, then died.

The automatic rollback began, but the deployment declarations assumed an older production-file version.

Recovery needs proof

- Rollback script runs against production-like machines
- Control plane remains usable under failure
- Old route actually serves users again
- Recovery state is independently verified

> The postmortem lesson: rollback scripts, infrastructure declarations, and release controls deserve a higher quality bar than feature code.

A successful deployment command is not proof that users recovered.

DATA CONTRACTS

Put deterministic boundaries around the uncertain model.

Input contract

Required fields, formats, data classes, size limits, and missing-data behavior.

Prompt contract

Allowed context, redaction rules, required policy, and source metadata.

Tool contract

Arguments, types, permissions, idempotency, confirmation, and error handling.

Output and log contract

Schema, citations, refusal behavior, safe formatting, and privacy-aware evidence.

Contracts do not remove model uncertainty. They make the surrounding system measurable, debuggable, and safer to operate.

PRODUCTION TRACE MINING

Real users reveal the cases your test plan did not imagine.

Sample

Randomly measure ordinary quality. Target low ratings, retries, refusals, long sessions, and escalations.

Cluster

Find failure patterns: stale context, bad tool call, missing policy, unsafe refusal, or overflow.

Promote

Sanitize high-value cases into regression tests with owners, slices, and release gates.

> Keep raw traces separate from sanitized eval cases. Production evidence can contain personal data, source code, secrets, and proprietary workflows.

Synthetic data explores the map. Production sampling shows where users actually walk.

ECONOMICS IS QUALITY

Choose the route that creates the most trustworthy value.

Quality

Does the route solve the real task across slices and repeated runs?

Latency

First token, full response, p95/p99, queueing, retry, and tool delay.

Cost

Tokens, retrieval, tools, judges, cache misses, human review, and failures.

Continuity

Privacy posture, region, provider availability, fallback, and rollback.

> A cheaper model that creates more escalations is not cheaper. A faster model that causes more refunds is not faster in business terms.

Test the router too. The right model route is part of the answer.

PUT IT INTO PRACTICE

Operate AI as a measurable system.

AI quality needs more than a model score: production traces, evidence-aware evaluation, rollout controls, data contracts, and economic discipline. IcebergQA helps teams make that operating model real.

Talk to IcebergQA