The Art of Reliable AI Agents

Foundation • Control • Trust

DANEYAL ANIS
AVP, DATA SCIENCE, COE, SUN LIFE FINANCIAL
CDAO TORONTO • MARCH 25–26, 2026

Why most agent pilots stall

The gap is not model capability. It is production reliability.

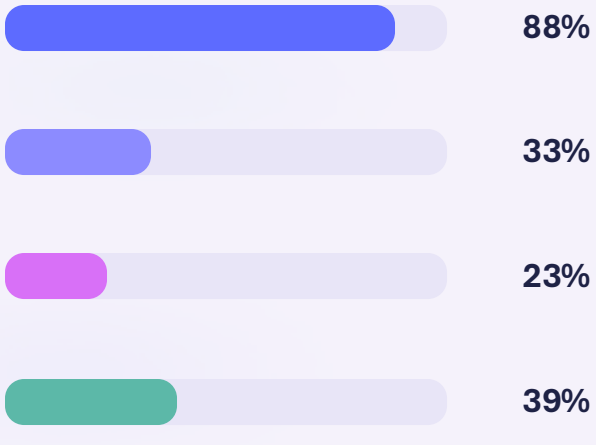
The production gap

- AI use is broad across enterprises
- Scaling is still early
- Agent success depends on workflow redesign, human validation, and engineering discipline

Reliable agents win when the surrounding system can fail safely, recover cleanly, and prove what happened.

McKinsey Global Survey 2025

Adoption is ahead of scale

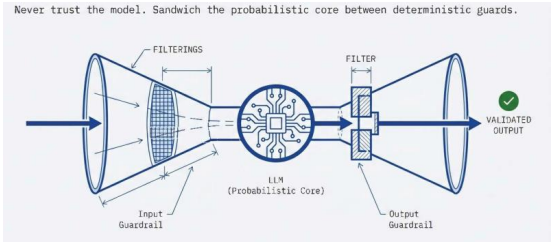


3 Core Operating Principles for Building Reliable Agents

Foundation

Loop • Probabilistic Compute • Schema • Persistence

Makes the agent executable as a system, not a prompt.



Control

Thoughts • Bowtie • Circuit Breaker • Collaboration

Constrains failure modes before they reach users or systems.

Trust

Frugality • Grounding • Evals • Standards

Makes the output useful, affordable, and auditable in production.

Foundation: engineer the loop

An agent is a stateful loop running on probabilistic compute.

Loop

The agent controls the next step. Design for multi-step state, not one-shot prompts.

Probabilistic compute

Outputs vary. Treat generation as an uncertain computation that needs checks.

Schema

Typed tool contracts keep the model from touching APIs loosely.

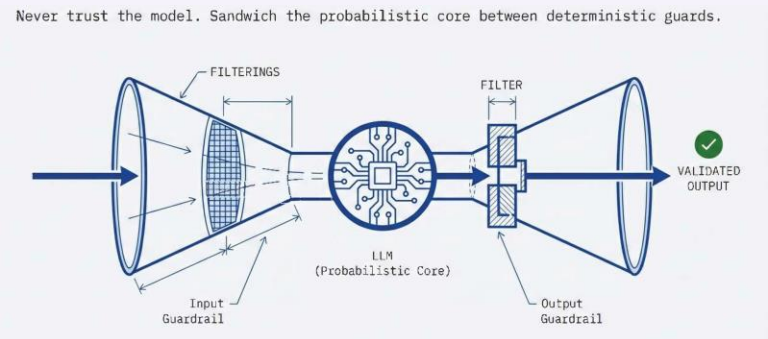
Persistence

Checkpoint state so the agent survives restarts, delays, and approvals.

These four laws turn prompting into systems design.

Control: constrain the failure modes

Reliable agents are wrapped in deterministic controls before they touch users or systems.



The model should sit inside a bowtie: intent comes in through guardrails, actions go out through validation.

Thoughts

Plan, reflect, then act on the stronger path.

Bowtie

Input guardrails and output checks around the model.

Circuit breaker

Stop loops, retries, and runaway spend from the outside.

Collaboration

Interrupt for human approval when confidence or stakes demand it.

Trust: useful, affordable, auditable

Production agents must control cost, stay grounded, pass evals, and fit open standards.

Frugality

Route simple work to cheaper models. Compress context and cache aggressively.

Evals

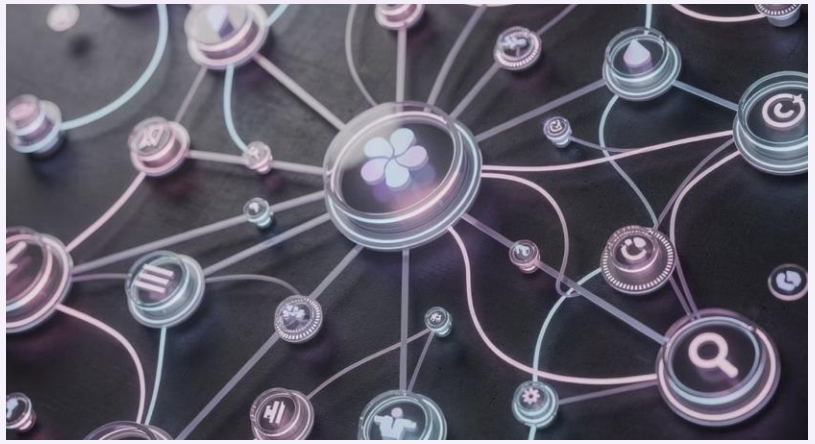
Use a golden set and regression checks before every release.

Grounding

Retrieve, rerank, and require evidence before making claims.

Standards

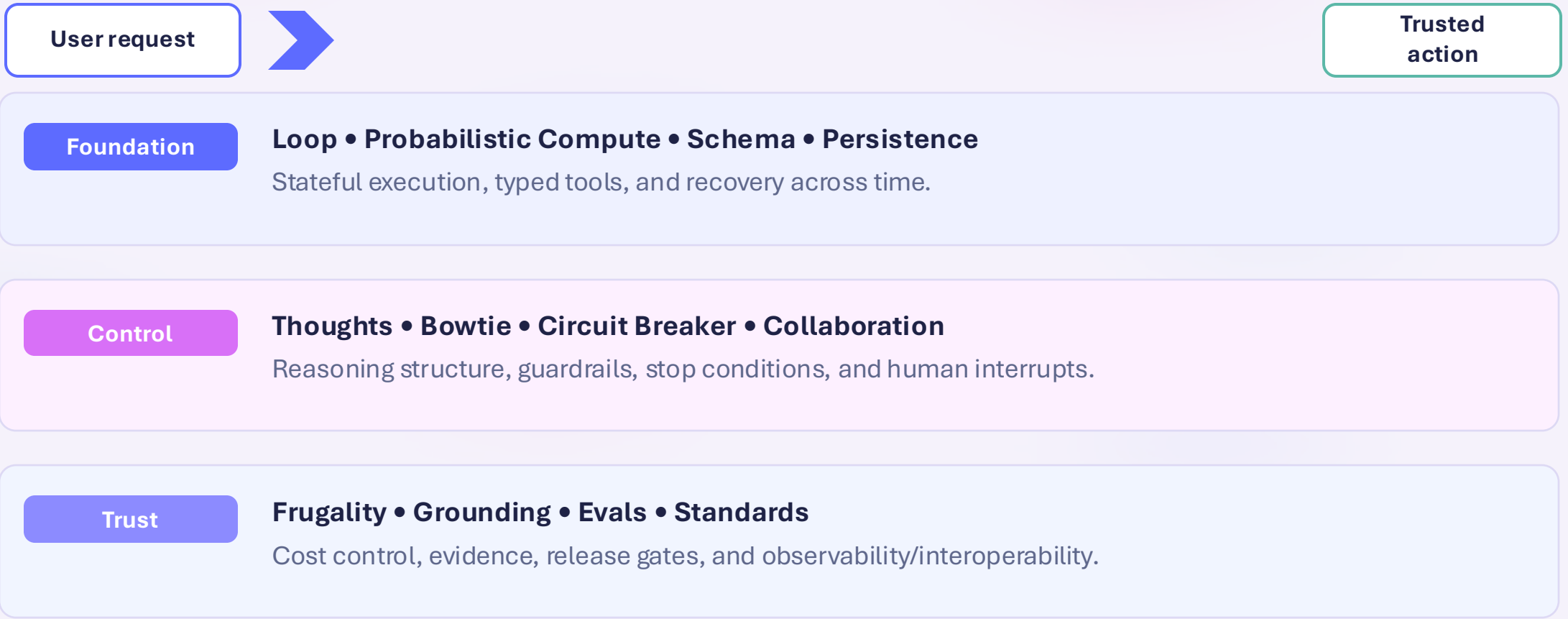
Use MCP for connectivity and OpenTelemetry for end-to-end traces.



Grounded answers are a product requirement, not a nice-to-have.

So, in summary...

Foundation makes the agent executable, Control makes it safe, Trust makes it shippable.



If you remember only three things...

1. Build the loop before you optimize the prompt
2. Wrap probabilistic models in deterministic control
3. Treat grounding and evals as release gates

Q&A