WHITE PAPER

Agents that act, *not agents that summarize.*

A summary is a demo. An agent earns its place when it can take an action inside a system of record and be held to the result.

Riverton & Associates, Inc. · 6 April 2026

Executive summary

Most systems currently described as agents retrieve information, interpret it, and produce text. A person then reads that text and does the actual work. That is a useful assistant. It is not an agent, and the distinction is not pedantic — it is the difference between a system that demonstrates well and a system that changes a number somebody is accountable for.

> The bar we hold to is specific: **an agent takes an action inside a system of record, and somebody can be held to the outcome.** Everything difficult about building one lives on the far side of that line, and nothing on the near side prepares you for it.

This paper is about what changes when a system stops describing the world and starts altering it: authority, reversibility, partial failure, audit, and the competence to decline.

Why summaries demo so well

A summarization demo is impressive in almost exact inverse proportion to what is at stake.

Nothing happened. No state changed, so there is nothing to reconcile, nothing to roll back, nobody to notify, and no audit trail to produce. The failure mode is a paragraph somebody disagrees with. The demo is safe because it is inert.

There is a second, subtler reason these demos flatter: summarization has no ground truth. A wrong summary looks exactly like a right one. It is fluent, plausible, and nobody in the room has read the underlying two hundred pages closely enough to catch the omission.

Compare that to an agent that issues a credit note for the wrong amount — that failure announces itself, to someone, in a way nobody can talk around.

This is why pilots stall. The demo cleared a bar that had been set low by the absence of consequences, and the production version has to clear a different bar entirely.

What changes the moment it acts

Everything the demo omitted arrives at once, and none of it is model work.

- **Authority.** On whose behalf is this acting, under what delegated limit? An agent inherits a permission set, and if that set is "whatever the service account can do", the blast radius is the whole system.
- **Idempotency.** Networks retry. Queues redeliver. Somebody clicks twice. If the action is not idempotent, the second execution creates a duplicate payment, ticket or order — and it will happen.
- **Reversibility.** Can this be undone, by whom, and within what window? Actions divide sharply into reversible and not, and that division should drive how much autonomy each is given.
- **Partial failure.** It completed steps one through three of five and then the API returned a 500. The world is now in a state no one designed. Multi-step actions need explicit compensation, or they need to not be multi-step.
- **Audit.** Months later somebody asks why this record was changed. The answer has to be specific: which action, on what evidence, under which rule, at whose authority. "The model decided to" does not survive contact with a regulator or a customer.
- **Reconciliation.** The system of record now disagrees with three other systems that were not told. Something has to close that gap.

None of these are improved by a better model. They are systems engineering, and they are where the actual work is.

What "held to the result" requires

The second half of the claim is the harder one. Accountability is what separates an agent from a suggestion engine, and it has concrete prerequisites.

- **A named owner.** A person accountable for the decisions the agent makes, in the same way they would be for a team member's. If nobody owns the outcome, nobody will investigate a bad one.
- **A measurable outcome, in business terms.** Not "hours saved" and not "queries answered" — the number the process exists to move. Activity metrics are how agent programmes avoid being evaluated.
- **A justification attached to every action.** Not a log line saying what happened, but a record of why: the evidence considered, the rule or threshold applied, the confidence.
- **An escalation path.** Somewhere for the agent to go when it should not proceed, staffed by someone who will actually look.
- **A control group.** Otherwise any change in the outcome can be attributed to the agent, and it usually will be.

If those five are not in place, the system may still be useful — but nobody is being held to anything, and its value is an assertion rather than a measurement.

The competence boundary

The most important capability an acting agent has is knowing when not to act.

An agent that acts on everything is more dangerous than one that acts on sixty per cent and escalates the rest, because the sixty per cent it handles well tells you nothing about the forty per cent it should never have touched. Coverage is a poor target; **coverage at an acceptable error rate** is the real one.

That requires a calibrated sense of its own confidence, an explicit threshold below which it declines, and an escalation treated as a correct outcome rather than a failure. Teams that measure "percentage handled autonomously" will tune that threshold in exactly the wrong direction.

The related discipline is the cost asymmetry: how much worse is acting wrongly than not acting at all? For a reversible, low-value action the answer may be "barely". For a payment, a customer communication, or anything touching a regulated record, the answer is "enormously" — and the threshold should reflect that rather than a default.

The decision does not belong inside the model

The architecture that survives production separates interpretation from execution.

The model interprets. It reads the messy input, works out what is being asked, and proposes an action. This is what it is genuinely good at.

A deterministic layer executes. Actions are explicit, typed, individually permissioned tools — not free-form access to an API surface. Preconditions are checked in code before anything runs. Limits are enforced outside the model, where they cannot be argued with. Every execution writes its own audit record.

> The model proposes; the deterministic layer disposes. This is the same division that makes expert systems durable, arriving in a modern context: the component that decides is inspectable, testable, and unable to invent a capability it does not have.

Practically, that means an agent should not have credentials. It should have a small set of narrow tools that have credentials, each of which validates its own preconditions and refuses outside them.

How to start

- **Pick one action, not a workflow.** Workflows are where multi-step partial failure lives. A single, well-chosen action delivers value and teaches you what your systems actually do under automation.

- **Run it in shadow first.** The agent proposes, a person executes, and you measure agreement. This gives a real precision number before anything is at risk, and surfaces the cases nobody anticipated.
- **Graduate on a bounded slice.** Let it act autonomously where the action is reversible and the value is low. Keep the caps low enough that a bad week is an inconvenience.
- **Widen by evidence.** Expand scope when the reversal rate justifies it, not when the roadmap says so. This is the discipline most programmes lack.
- **Keep a human where the asymmetry demands one.** Not everywhere, and not nowhere — where the cost of acting wrongly is much larger than the cost of waiting.

How to tell whether it is working

- **Reversal rate.** The share of autonomous actions later corrected or undone. The precision measure that matters, and the one to report.
- **Escalation rate, and its trend.** Falling escalation with a stable reversal rate is genuine progress. Falling escalation with a rising reversal rate is a threshold set too loosely.
- **Time to detect a bad action.** If it is measured in months, the audit trail is not doing its job.
- **The business outcome against the control group.** The only measure that answers the question the project was funded to answer.
- **Not** actions taken, queries answered, or hours notionally saved. Those go up whether or not anything improved.

Conclusion

A summary is a demo. It is genuinely useful, it is easy to build, and it commits to nothing — which is exactly why it clears its bar so comfortably, and why so many of them never become anything more.

An agent earns its place when it changes something real: a record altered, a payment made, a case closed, a customer told. Everything hard about that is downstream of the moment it acts — authority, idempotency, reversal, audit, and the judgement to stop. None of it is model work, and all of it is the work.

Build the second thing. Then find out whether it was right, from a number, against a control, with somebody's name on it.

Riverton & Associates researches, designs, develops and implements production software and AI systems that take actions and are measured on them.

`Riverton & Associates, Inc. · Austin, Texas · riverton-assoc.com`