RJL SOLUTIONS

EXECUTIVE GUIDE / STRATEGY THROUGH PRODUCTION

The Connected Enterprise Data & AI Blueprint

An executive guide to the architecture and operating decisions that move enterprise AI into production.

ARCHITECTURE · OPERATING MODEL · FIRST CAPABILITY

Richard White
RJL Solutions · rjlsol.com

OCTOBER 2026
EDITION ONE

THE EXECUTIVE DECISION

Enterprise AI becomes a capability when meaning, access and ownership are settled.

A capable model and a modern data platform help. They don't settle those operating decisions. Enterprise AI becomes a business capability when the information it uses has agreed meaning, the access it gets is controlled and the work it performs has an accountable owner.

This guide explains the connected enterprise foundation: governed data products, shared business meaning, traceable knowledge, controlled access and orchestration, and trust built into operation. It's for executives responsible for technology, data, AI, security and the processes they support.

Start with one valuable workflow, delivered as a complete governed slice. Prove it can use real data, respect real permissions, give useful answers and run at acceptable cost. Expand once the evidence supports it. A 90-day pattern and a readiness diagnostic turn the architecture into a decision you can make now.

How to use this guide

Read the first three sections to frame the decision. Work through the architecture with the delivery team. The 90-day pattern turns it into a first release. Complete the diagnostic together, then agree the next action on the worksheet.

INSIDE THE GUIDE	PAGE
Why AI stalls, and how to choose the first workflow	2–3
The connected enterprise foundation	4
Five capability areas and the four-part system	5–10
An illustrative claims support workflow	11
Organization, operation, and transfer	12–14
Evaluation and production economics	15
A 90-day starting pattern and release decisions	16–17
The 10-question production-readiness diagnostic	18–19
First capability worksheet	20
Working session and references	21

THE DECISION TO MAKE — Authorize a bounded first capability. Give it a named business owner, agreed evidence requirements and an explicit decision at the end: proceed, reshape or stop.

Why enterprise AI stalls after the demo.

A demonstration usually answers a narrow question under favorable conditions. The people testing it know the data. Someone curated the documents. The service account has broad access. And the workflow stops before any consequential action.

Those conditions hide the constraints that will decide whether the capability works in daily operations. Once the audience grows, the same word may mean different things across business units. A document may be current for one product and obsolete for another. A model may produce a persuasive answer without the evidence to justify it. An agent may hold technical access that exceeds the authority of the person asking.

The problems live between the platforms

Unresolved meaning. A warehouse can store customer, policy and revenue records without deciding which business definition applies to a particular question. When those decisions are missing, an AI system has room to invent the calculation or pick the wrong source.

Knowledge without provenance. A search index can find a relevant paragraph without establishing whether the reader may see it, whether the policy is still in force or whether it applies to the case at hand.

Access without delegation. Handing an assistant a service credential can make a prototype work fast. Production needs a defined relationship between the caller, the agent, the requested purpose and the permissions the source enforces.

Delivery without operation. A pilot can succeed without proving recovery, incident handling, change control or ongoing evaluation. All of these belong to the capability a business owner inherits.

Diagnose the blocked seam

Ask the team to trace one recent request from business intent to source data, policy decision, answer and evidence. Find where the trace turns into an assumption. That beats choosing another model or adding another interface as a starting point.

The first release should close those gaps for its own workflow. It doesn’t need to solve every definition or connect every source across the enterprise. It does need to show the complete pattern within its approved scope.

Choose a workflow with a measurable outcome.

Begin with work that matters to an accountable business owner, and that you can observe before and after the change. Good candidates have a repeated decision or information task, identifiable users, reachable source material and a boundary narrow enough to test.

A claims support assistant, a finance reconciliation workflow or a controlled internal knowledge service can each be a starting point. Map the work as it happens today. Record who gathers information, who reconciles it, who reviews it, where the waiting happens and what follows when the answer is wrong. That shows which activity AI could improve, and which needs a data or process change first.

Define the economics before the build

Use a baseline period that captures normal workload, exceptions, case mix and seasonality. Count review and exception handling in future operating cost.

MEASURE	WHAT TO ESTABLISH
Business outcome	Cycle time, rework, reconciliation effort, or another workflow result
Volume and adoption	Eligible transactions, actual usage, and completed work
Quality and risk	Material errors, escalation rate, prohibited disclosures, and policy breaches
Full operating cost	Model, compute, retrieval, support, review, and recurring data maintenance
Reuse	Which certified products or controls the next workflow can consume

For planning, estimate annual capacity value as eligible volume × adoption × net minutes saved per case, after added review and correction, × loaded hourly cost ÷ 60, less incremental operating cost. For example, 120,000 cases, 60 percent adoption and 6 net minutes give 7,200 hours; at $65 an hour that is $468,000, or $248,000 net of $220,000 in annual operating cost. In the cash view, only the 2,000 contractor hours the owner plans to stop buying count: $130,000 against the same $220,000 is a net cost of $90,000. The views are not additive and exclude implementation cost. Illustrative figures; time becomes cash only when staffing or purchased capacity changes.

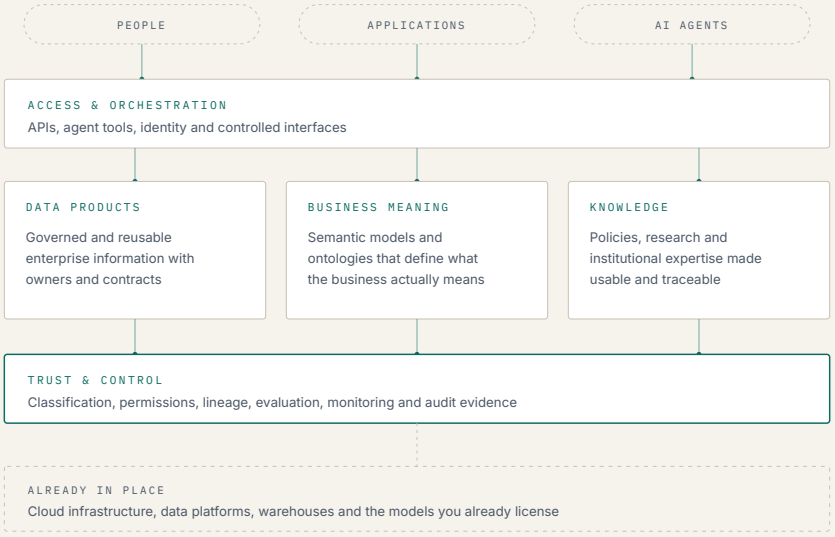
Choose the first scope deliberately

Prefer a candidate with enough value and a workable route through data, identity and review. Executive visibility and demo quality alone are poor reasons to choose. If a valuable workflow needs unresolved identity matching, unreachable documents and autonomous actions all at once, start with a narrower information task inside it.

FRAME DECISION — Proceed when the sponsor accepts the baseline, the owner can change the workflow and the deciding constraints are testable; otherwise reshape or stop.

The connected enterprise foundation.

The foundation connects the platforms you already own to the information and actions the business needs. Five capability areas carry distinct responsibilities. You implement them together for each release, with trust and control applied throughout.



Five capability areas sitting above the platforms already in place, with trust and control as the substrate beneath all of them.

What the diagram means

People, applications and agents reach enterprise capabilities through controlled interfaces. Those interfaces expose governed data products, agreed business definitions and curated knowledge. Identity, permissions, lineage, evaluation and monitoring govern every route.

One request may use more than one capability. A claims question can need a current status from an operational API, a benchmark from a certified metric and a procedure from a document corpus. Orchestration coordinates those paths without granting any extra authority.

Existing warehouses, cloud infrastructure and operational systems stay relevant. The design adds explicit contracts above them. The first release should reuse mature services where they deliver the required behavior, and isolate the gaps that need new engineering.

ARCHITECTURE TEST — Can the team explain what each component owns, where each control is enforced, and what the system does when that component or control fails?

Data products, and continuing ownership.

A data product is a governed unit of information or capability designed for repeated use. It has a stable identity, a responsible owner, an approved interface and a contract that explains what a consumer can rely on. A table becomes useful as a product when someone maintains those commitments through its lifecycle.

Publish the contract a consumer needs

The product contract explains the business purpose, record grain, authoritative sources, important definitions, allowed joins and known limitations. It also states freshness and quality expectations, access restrictions, lineage, support arrangements, cost limits and how changes get communicated.

A manifest such as `data-product.yaml` can hold these declarations in a versioned form, and is useful when deployment, testing, catalog publication and monitoring all use it. A completed manifest without working enforcement or support is only documentation.

Purpose and lineage are independent classifications. A product can be operational and foundational at once.

PURPOSE OR LINEAGE	TYPICAL INTERFACE	CONSUMER EXPECTATION
Analytical	Certified table, view, or metric service	Defined grain, reproducible calculations, published freshness
Operational	Bounded system-of-record API or event	Current business state and a documented service contract
Foundational	Source-aligned product	Owned close to the domain that produces the information
Derived	Joined or reshaped product	Owns its additional logic and inherited dependencies

Keep certification current

Move the product through its delivery stages, from proposed to retired, and record certification states separately. Assign one owner accountable across those stages. Specialists contribute reviews and engineering; the consumer can always see who is accountable.

When a source changes, assess the effect on definitions and consumers. When quality or freshness falls below the contract, expose the degraded state and apply the agreed restriction. When you replace a product, give consumers a migration window and retire the old interface on purpose.

For the first workflow, certify only the dependencies it needs. Three to five products may be a useful planning range. Readiness depends on meeting their contracts. A count proves nothing.

EXECUTIVE IMPLICATION — Fund continuing ownership and maintenance alongside delivery. Reuse depends on a product staying trustworthy after its original project has ended.

05 / CAPABILITY AREA TWO

Business meaning keeps answers consistent.

The question “how many customers do we have?” is a definition problem before it’s a query problem. For an insurer, the customer might be a policyholder, an insured person, an employer holding a group contract or a household. Several counts can be valid for different purposes. A governed foundation makes the chosen purpose and definition explicit.

Separate calculations from relationships

A semantic model defines metrics, dimensions, time rules, grain and permitted calculations. It can fix how earned premium or average time to pay gets computed, and which exclusions apply. Verified questions and expected results help test whether a conversational interface uses that model correctly.

An ontology defines business entities and their relationships. It can tell a policyholder from an insured person, describe how a claim relates to a policy and keep relationships across product lines. Use graph capabilities where the workflow needs to traverse relationships or reason about hierarchy.

NEED	GOVERNED REPRESENTATION	EVIDENCE TO RETAIN
A total, rate, or trend	Approved metric and dimensions	Definition, version, query, and source snapshot or as-of time
Parties connected to a claim	Approved entities and relationships	Source identifiers, mapping rules, and relationship provenance
A term with several valid meanings	Purpose-specific definition	Selected context and the authority that approved it

Agree on meaning before generating it

AI can propose definitions from schemas, policies and existing documentation. Domain experts have to decide the business meaning, including exceptions no schema shows. Generated mappings and views then need validation against the physical source, and regression tests against approved examples.

Version changes with their effective dates. A revised definition may legitimately change an answer. The team still has to be able to reproduce the answer under the prior definition, so keep the historical context that reconstruction needs.

Don’t force every business unit into one definition where different product or regulatory contexts call for different measures. Govern the differences and their intended use. Consistency means the same question, in the same context, uses the same approved logic.

EXECUTIVE IMPLICATION Assign a business authority to settle meaning. Engineering can encode a decision. It can’t supply the business mandate to make one.

06 / CAPABILITY AREA THREE

Knowledge needs provenance and permissions.

Much of the information enterprise work needs lives in policies, procedures, correspondence and case files. Making that material searchable creates a new route to it. It also creates obligations: keep the document's authority, scope, freshness and permissions intact as it gets parsed and indexed.

Curate the corpus around the workflow

Choose the sources that can answer the selected users' questions. Identify the source owner, effective date, document type, product or jurisdiction, access rules and retention conditions. Establish how a replacement, deletion or permission change reaches the index and any relevant caches.

Parsing creates chunks, embeddings and indexes. Each derived artifact needs enough metadata to apply the original restrictions and return traceable evidence. Test the result with a user who should see the document and a user who shouldn't. Filtering only the final answer comes too late if restricted content has already entered the model context.

Ground an answer in the right source

For an answer about a procedure, return citations that let the user inspect the supporting passage and its document context. Keep policy text distinct from the model's explanation. If sources conflict, or the applicable version is unclear, ask for context or escalate to the designated owner.

A retrieved document is evidence. Instructions embedded in it must never change the assistant's permissions, tool choices or operating policy. Put hostile and misleading document content in the evaluation pack, so you test this boundary instead of assuming it.

CORPUS CONDITION	REQUIRED OPERATING BEHAVIOR
Permission revoked	Remove access from retrieval and affected caches within the agreed window
Policy superseded	Prefer the applicable approved version and preserve historical provenance
Source deleted or expired	Remove derived artifacts according to the retention design
Index stale	Disclose freshness or restrict the affected workflow
Evidence incomplete	Constrain the answer, clarify, or route to review

EXECUTIVE IMPLICATION Knowledge retrieval needs content ownership and operational upkeep. A bigger corpus helps only when the team can explain which material is authoritative and who may use it.

07 / CAPABILITY AREA FOUR

Access and orchestration preserve authority.

An enterprise interface should make approved capabilities easier to use while keeping the source's controls. It might be an API, an application or a named agent tool. Its contract should define the allowed operation and make prohibited behavior testable.

Identify both the caller and the agent

For a delegated request, record the human or application starting the work and the agent acting for it. Three things bound effective access: the caller's entitlement, the approved scope of the agent or tool, and any purpose or context restrictions. Use scoped workload identities where services need their own authority.

Propagation has to reach the point of enforcement. A gateway accepting a user identity doesn't prove that a downstream database query runs with the right restrictions. Test the entire path, including pooled connections, caches, federated sources and asynchronous work.

Route by the work requested

Send metric questions to approved semantic interfaces, relationship questions to approved entity mappings, policy questions to curated retrieval and live status requests to the right system of record. Split mixed requests into a constrained plan. When intent or applicable context is uncertain, clarify.

Expose narrow tools such as `get_claim_status` rather than broad `run_sql` or `update_record` operations. An agent-facing protocol, including the Model Context Protocol (MCP), makes a tool callable. It does not establish that the tool is safe, governed or correct.

Certify actions separately

A read entitlement confers no authority to change a claim, approve access or contact a customer. A command path needs a named business operation, typed inputs, validation, authorization, approval where required, duplicate protection and an observable final outcome.

For asynchronous writes, record whether a command was accepted, applied, rejected or left unresolved. Use durable command state and a controlled remediation path. An idempotency key helps prevent duplicate effects during retries. It can't establish that the requested action was right for the business.

EXECUTIVE IMPLICATION

Keep the first release read-only unless a separately justified command capability is essential. Add actions once ownership, approval, failure handling and recovery are ready to carry the consequences.

Trust and control are part of every request.

Trust depends on controls that work where information is read or an action is applied. Classification helps identify sensitive content. Authorization decides who may use it, and for what. Masking, filtering and service controls enforce the decision. Evidence records what happened.

Keep intelligence and enforcement distinct

Data security posture management can discover sensitive information and exposure. A catalog can describe it. Neither one guarantees that a warehouse, document index, API or agent tool enforces the required policy. Each route needs a documented enforcement point and a test of its behavior.

Semantic metadata guides how a question gets interpreted. Never rely on it as the only access boundary. Apply controls at the storage or source service, and at other relevant interfaces. Confirm the actual execution path for the technology you chose.

CONTROL	AUTHORITY TO ESTABLISH	BEHAVIOR TO VERIFY
Identity and delegation	Identity and workload owners	Unknown or out-of-scope callers are denied
Classification and use	Security, privacy, and domain owners	Sensitive data is restricted across every consumption path
Tenant isolation	Service and security owners	One tenant (a business unit, legal entity or client boundary) cannot retrieve another’s information
Policy propagation	Named integration owner	Changes are deployed, reconciled, and rolled back safely
Audit evidence	Operating and control owners	A request can be reconstructed without unnecessary sensitive logging

Design for uncertainty

Declare the authoritative policy source, its enforcement targets, the allowed propagation delay and the reconciliation process. Decide what happens when a policy update fails or the current state is uncertain. For sensitive access, the right default is denial, or a deliberately restricted mode.

Use a durable correlation identifier to connect the request, delegated identities, policy result, source access, component versions and the returned response or command outcome. Store the evidence under its own access and retention controls. Don’t copy raw sensitive material into logs just to make debugging convenient.

EXECUTIVE IMPLICATION — Require evidence of enforcement across the complete path. A policy document or a configuration screenshot can’t show how a production request actually behaved.

09 / A SECOND VIEW

The four-part system.

The Brain and Operating Layer reference architecture uses four parts to explain how the system works together. They're a second view of the same foundation. They aren't four independent programs to fund and integrate later.

PART	RESPONSIBILITY	CONNECTION TO THE FIVE CAPABILITY AREAS
Brain	Interpret business entities, metrics, and knowledge	Business meaning and knowledge applied to governed products
Nervous system	Carry requests through approved APIs and tools	Access and orchestration
Body	Provision environments and support delivery and operation	Product lifecycle, self-service, workflow integration, and support
Immune system	Detect exposure and sustain security controls	Trust and control across every route

The Brain has five conceptual layers

Physical information supports approved entity and relationship metadata. That metadata can generate tested views over the sources. Semantic models then define the measures the business uses, and orchestration picks the right route for each request.

These layers describe responsibilities. Nobody has to buy five products. A first implementation may cover several responsibilities with existing services. Build the ontology or graph path only where the selected workflow needs it. Whatever components you choose, keep meaning and access enforcement distinct.

The Operating Layer makes the foundation usable

The Operating Layer is the operating system around the Brain: the provisioning and operating pattern that makes the foundation usable. It includes environments, approved tools, access workflows, self-service and lifecycle controls. It's a pattern in the reference architecture. You can't buy it.

Self-service can progress in three steps. A sandbox uses synthetic or appropriately de-identified information. A managed environment adds approved real data. A certified production capability comes last. Each promotion changes the obligations. A useful experiment still needs tested contracts, support and approval before others rely on it.

The control plane connects the evidence

The data plane carries queries, retrievals and commands. The control plane supplies identity, authority, classification, approvals, versions and evidence. Join the two for each request. Then the team can explain both the result and the conditions under which producing it was permitted.

EXECUTIVE IMPLICATION — Choose components by their bounded responsibilities and exit conditions. When a platform or model changes, keep the contracts and control behavior intact.

An illustrative claims support workflow.

Take a multi-line life and disability insurer with separate policy, claims and group benefits systems. It wants to help an examiner assemble case context and find the applicable procedure. This example illustrates the architecture. It isn't a client deployment, and it claims no achieved results.

The examiner asks: "Who is connected to this claim, what is its current status, and which procedure applies?" The first release supports investigation and information gathering. The examiner keeps responsibility for the case decision. Claim adjudication, payment authorization and external communication stay outside this release's scope.

Trace the request through the foundation

01 **Establish authority.** Authenticate the examiner, identify the assistant, and evaluate the allowed claim population and purpose. Give the assistant a constrained set of read tools.

02 **Resolve the entities.** Use approved identifiers and relationship rules to link the claim, insured person, policyholder, policy and employer where they apply. Show unresolved or conflicting matches. Never treat them as certain.

03 **Read current state.** Get status from the designated source API, or from a certified product whose freshness suits the task. Keep the source and the as-of time.

04 **Retrieve applicable knowledge.** Search the curated procedure corpus using product, jurisdiction, effective date and access restrictions. Return the supporting passages, with citations.

05 **Assemble the answer.** Keep source facts distinct from interpretation. Explain material gaps, and route ambiguity to review. If the user also asks for a benchmark, use the approved metric and disclose its definition and period.

06 **Retain the trace.** Link the caller, delegated identity, product and document versions, policy decision, tool calls and response under one correlation identifier. Apply the agreed retention and redaction rules.

What would demonstrate value

Measure time spent assembling context against the baseline, using comparable cases. Track actual use, correction rate, unresolved matches, unsupported answers, prohibited disclosures and cost per completed task. Does the assistant reduce work once review and exceptions are counted? Find out.

Before expansion, test a revoked user, an inaccessible document, a conflicting identity match, a stale status source and a hostile instruction embedded in a file. Define the expected result for each. A plausible answer in the normal case isn't enough evidence for release.

Organizing domains and shared teams.

The first capability needs an organization that can resolve meaning, authorize use, deliver changes and keep the service running. You don't need an enterprise reorganization before work can start. You need clear decision rights, and a delivery team with access to the people who hold them.

Give the domain authority over its work

Business domains own the process, its definitions and the foundational information they produce. A cross-functional product team combines the domain owner with data engineering, integration, AI, architecture and the relevant control specialists. Its remit covers delivery and continuing operation. A project milestone is only part of it.

Shared platform and enablement teams maintain reusable identity, deployment, catalog, observability and evaluation services. Specialist disciplines, such as data engineering and security engineering, maintain engineering standards. Their members can work in domain teams while using common practices and components. Don't recreate those services in every business unit.

LEADERSHIP ROLE	AUTHORITY THE FIRST CAPABILITY NEEDS
Business process owner	Workflow changes, outcome acceptance, and adoption
Data leadership	Definition authority, data-product standards, and stewardship
Security leadership	Security policy, enforcement standards, and incident coordination
Risk, model risk and compliance	Independent review or validation, risk acceptance, and regulatory mapping
Technology leadership	Platform reliability, integration, and engineering support
AI or digital product leadership	Agent scope, evaluation, product behavior, and delivery priorities

Titles differ by enterprise. Name the actual decision holders and their delegates. Data, security, technology and AI leaders need shared accountability for the complete capability.

Resolve disagreements through explicit decision rights

Document who approves a definition, who grants access, who accepts residual risk, who can stop a release and who funds ongoing maintenance. Set an escalation route and a decision time frame. The product owner coordinates those decisions without taking authority that belongs to a control owner.

Start with a small named team. Add specialists for the workflow's constraints, such as knowledge engineering for complex entity relationships or privacy expertise for sensitive content. Measure review bottlenecks before adding committees.

EXECUTIVE IMPLICATION — Use the first capability to prove the working relationships. Expand the organization and skills structure once the evidence shows which responsibilities need lasting capacity.

Ownership and internal transfer.

The operating model connects business authority with technical accountability. Name who accepts the outcome, who maintains the capability and who runs it after delivery.

Name the people who can decide

Appoint a business process owner, and a product owner responsible for the capability's lifecycle. Establish definition, access and service ownership. In a small team one person may hold several roles. Each responsibility still has to stay explicit.

RESPONSIBILITY	ACCOUNTABLE ROLE	DECISION OR EVIDENCE
Business result and workflow	Process owner	Baseline, target, adoption plan, and outcome review
Product lifecycle	Product owner	Scope, dependencies, versioning, and retirement
Meaning and quality	Domain definition authority	Approved definitions, exceptions, and reconciliations
Risk and permitted use	Risk, compliance and security owners	Access policy, risk conditions, independent review, and release approval
Reliable operation	Service owner	Objectives, monitoring, recovery, support, and incident response

Keep shared standards close to delivery

Use common identity, contract, logging and evaluation standards across domains. Keep business definitions and source responsibility with the domain that has the authority to maintain them.

Give reviewers concrete artifacts during the build: a request trace, a definition, a policy test, a recovery exercise. That helps settle disagreements while the design can still change. Approval based only on a final presentation leaves too much of the behavior unexamined.

Transfer begins in the first sprint

Internal engineers should work in the code, the tests and the operations from the start. Maintain runbooks alongside the implementation. Before you call transfer complete, have the receiving team demonstrate routine operation, access revocation, incident handling and rollback.

EXECUTIVE IMPLICATION — Close the authority gap early. A team accountable for a result needs the mandate to change the process, and the cooperation of the teams that control its dependencies.

Operating the capability.

Users rely on a production capability's answers, freshness and availability. Define how the team admits changes, handles failures, maintains controls, and restricts or retires the service.

Make promotion a deliberate change in obligations

A sandbox supports exploration within approved data limits. A managed sandbox supports controlled testing with real data. Production needs tested contracts, enforcement, support and acceptance for a stated scope. Tell users which stage they're using.

Version the products, documents, definitions, prompts, tools, models and policies that affect the outcome. Assess dependencies and required evaluation before release. For breaking changes, identify affected consumers and provide a migration or rollback path.

Establish a working review cadence

REVIEW	PARTICIPANTS	DECISION SUPPORTED
Daily, during early operation	Service and delivery owners	Incidents, freshness, cost anomalies, and immediate restrictions
Weekly product review	Process owner, product owner, technical lead	Adoption, exceptions, backlog, and workflow improvements
Before material release	Delivery and relevant control owners	Test evidence, changed scope, approval, and rollback
Monthly outcome and risk review	Business, service, and control owners	Benefit realization, control drift, investment, and expansion

Connect incident response to business impact

Define who can disable a tool, restrict a user population, roll back a model or suspend a product. Security incidents and incorrect-answer incidents may need different specialists. Both need a traceable owner and a way to tell users. Preserve evidence, contain the affected scope, fix the cause and add a regression case before restoring the behavior.

Maintain retention and deletion across source copies, retrieval indexes, caches and any agent memory. If you introduce memory, give it a separate purpose, access boundary, retention period and correction process. It must never become an ungoverned second store of enterprise information.

OPERATING REVIEW — Continue only while the capability's value, controls and support stay demonstrable. Shrink or retire unused or degraded functions. Don't carry them indefinitely as hidden dependencies.

Evaluation and production economics.

An available service can still give an incorrect or unauthorized answer. Production evaluation has to examine behavior, not just uptime. Build a versioned test pack for the actual workflow, with expected answers, expected refusals and explicit treatment of ambiguous cases.

Set release criteria by consequence

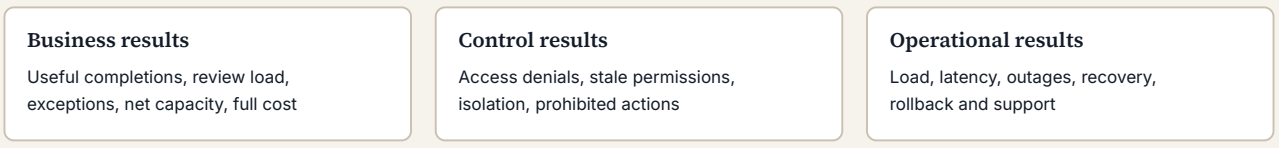
Deterministic calculations should reconcile to their approved definitions and sources. Language generation and retrieval need task-specific criteria, with thresholds chosen for the workflow's risk. A single average accuracy score can hide a serious access failure.

EVALUATION DIMENSION	EXAMPLE EVIDENCE
Meaning and calculation	Approved questions reproduce the expected metrics and joins
Knowledge grounding	Citations support the answer and use the applicable document
Identity and isolation	Unauthorized and cross-tenant requests are refused
Routing and ambiguity	Mixed requests are constrained; unclear intent triggers clarification
Tool and action behavior	Malformed inputs and unapproved side effects are blocked
Recovery and cost	Limits, dependency failures, retries, and rollback behave as designed

Run the pack before changes to models, prompts, definitions, tools, routing, policy or retrieval configuration, and before material source and corpus changes. An unresolved critical failure in disclosure, unauthorized action, data integrity or auditability blocks the affected release, however high the average quality score.

Release a named combination

A release is a named combination of versions: model, prompts, retrieval configuration and index, definitions, tools, policies and the workflow itself. Judge that one combination on business, control and operational evidence together. Keep the name, so you can identify, compare and roll back the evaluated capability.



Judge the same named release, a fixed combination of versions, on all three. A favorable average cannot override a failed critical control.

Exhibit: Release evidence covers the whole workflow.

Budget the full unit of work.

Attribute model, compute, retrieval and API costs to the product, workflow and calling identity. Include review effort, exceptions, support and recurring source upkeep. Track cost per completed useful task as well as cost per answer. Repeated attempts make those very different numbers.

Set limits on concurrency, retries, execution time, retrieval volume, context size and spend, and decide what the user sees when a limit is hit. Start with showback. Move to chargeback once the allocation model is agreed.

Publish availability, latency and freshness objectives, and test recovery and access revocation. Keep monitoring quality and business outcomes after launch. Changes in sources, user behavior and case mix can invalidate the original evidence.

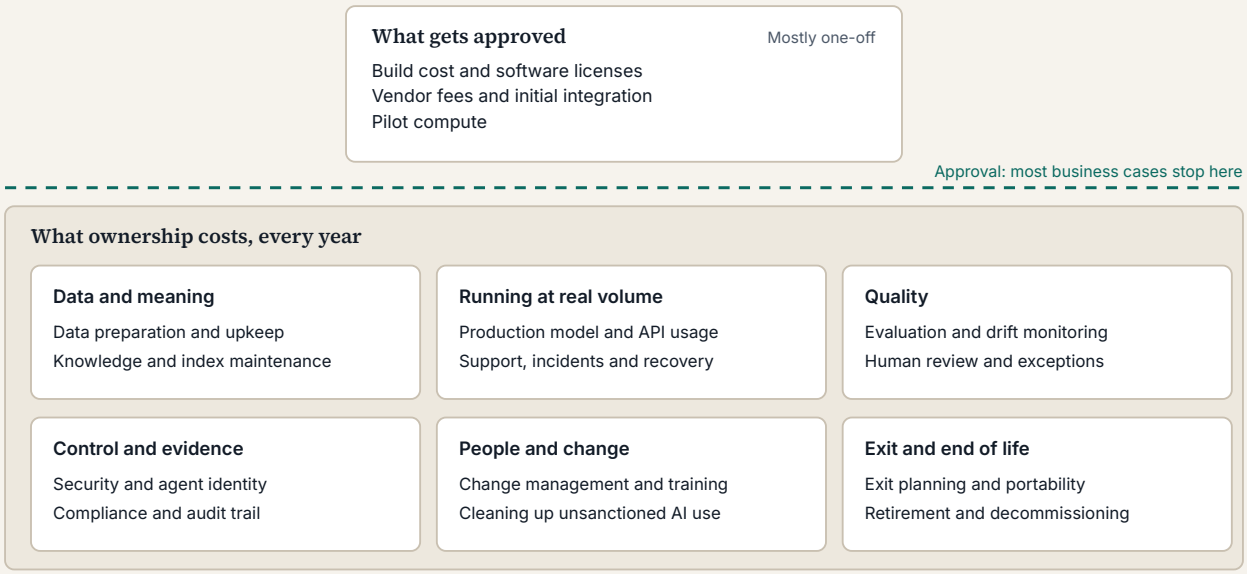


Exhibit: The business case approves the tip; ownership pays twelve recurring costs, every year.

RELEASE DECISION Expand when the capability passes its risk criteria, runs within its service and cost bounds and improves the intended work. Otherwise fix it, constrain it or stop.

A 90-day starting pattern.

The first 90 days should produce evidence about one governed capability, and a decision about what comes next. The sequence below is a planning pattern for a bounded workflow. Access lead times, procurement, source fixes and control reviews can all change the schedule. Day 90 is a review point. It isn't an automatic production deadline.

PERIOD	MAIN WORK	EVIDENCE AT THE BOUNDARY
Weeks 1 to 3	Frame the workflow, baseline, constraints, definitions and target design	Agreed business case, ownership, scope, risk conditions, and plan
Weeks 4 to 8	Prove an end-to-end slice with real data and controls in the enterprise environment	Working slice, evaluation results, request trace, and revised economics
Weeks 9 to 13	Begin Build: owned products and interfaces, release pipeline, controls and monitoring. Build continues past day 90, release by release	Certified dependencies, release pipeline, monitoring, limits, tested runbooks, and evidence for the actual release state
From first real use	Run controlled operating cycles and demonstrate internal ownership	Outcome readout, incidents and corrections, handover demonstration, and next decision

Keep the scope small enough to finish

Start with one workflow and a defined user group. Include the minimum certified data products, a thin set of approved metrics where needed and one curated corpus where retrieval is required. Add relationship modeling only for questions that need it. Use a small set of narrow read tools with complete identity and policy enforcement.

The output has to include working software and the evidence needed to operate it. Avoid a first release that connects many sources and can't explain the meaning, permissions and failure behavior of any complete request.

HOW THIS MAPS TO OUR PHASES

RJL's delivery phases are **Frame, Prove, Build, and Run & Transfer.** Frame typically takes two to three weeks and Prove four to six. Build typically takes eight to sixteen weeks per release, so it continues past day 90, and Run & Transfer begins when approved work reaches real users. This 13-week example uses five weeks for Prove and begins Build in week nine.

Make each phase end in a decision.

A phase boundary should decide whether the evidence justifies the next commitment. The business owner, delivery lead and relevant control owners should all inspect the same artifacts. Set the criteria during Frame, so a persuasive demonstration can't redefine success later.

Frame decides whether the problem is worth solving

Confirm the workflow owner, eligible users, baseline, expected value, material risks, reachable sources and internal operating destination. Settle who can approve meaning and permitted use. Continue when the problem is valuable and the deciding assumptions can be tested. Reshape or stop when the economics or dependencies don't support it.

Prove decides whether the slice earns further investment

Use real enterprise data in an approved test environment. Demonstrate the complete path for a representative task, including identity, source restrictions, citations or query provenance, and evidence. Test negative cases and dependency failures. Compare performance and cost with the baseline, and revise the design when the findings call for it.

Don't promote a slice just because a small group finds it useful. Check whether the evaluation covers the intended production population and the important exceptions. Any new population, source or tool may change the risk and the release conditions.

Build decides whether the capability is ready to rely on

Require owned dependencies, versioned contracts, appropriate approvals, passing evaluation, operating limits, monitoring, incident procedures and a tested rollback. Confirm the user workflow for incomplete evidence, denied access, stale data and an unavailable service.

If the evidence is incomplete, keep the capability within its approved testing scope, or release a narrower function. Never carry an unresolved critical failure into production through an unowned exception.

Run & Transfer decides who owns the next cycle

Operate with the internal team. Track benefit realization, errors, exceptions, user behavior and full cost. Have the receiving owners demonstrate support, change review, recovery and access revocation using their own procedures.

The day-90 review

At the 90-day review, choose one of three outcomes: proceed (expand, if a live release meets the release test, or continue while closing a named gap); reshape around the evidence; or stop. Record the reason, owner, funding and next review date.

TRANSFER CONDITION	The internal team sustains the service and its controls, and the business owner accepts the outcome and the continuing investment.

The 10-question readiness diagnostic.

Use this diagnostic for one named workflow. Bring its business owner, technical lead and control owner together. Answer with evidence from the implemented path. A roadmap doesn't count. Treat it as a discussion aid; it isn't a certification or a validated statistical assessment.

Record Yes, Partial, No or Unknown for each question. **Yes** means implemented and demonstrated for the proposed scope. **Partial** means a defined gap remains. **No** means it is absent. **Unknown** means the team cannot produce evidence. Add an evidence link, a gap owner and a target date on a working copy.

1 **Is there an owner and a measurable outcome?**

Can a named business owner change the workflow and accept its results? Show the baseline, target, eligible population and measurement method. Include review effort and exceptions in the comparison. A sponsor with no authority over the process doesn't close this gap.

RESPONSE _____ EVIDENCE OR GAP _____

2 **Are the information dependencies certified?**

Can the team identify the approved data products and their owners, interfaces, quality, freshness and lineage? Show current contracts, and evidence that the dependencies meet them. A catalog entry alone proves nothing about readiness.

RESPONSE _____ EVIDENCE OR GAP _____

3 **Is business meaning approved and tested?**

Who decides what customer, policy, revenue or the relevant metric means? Show versioned definitions, grain, important exceptions and tests against known answers. Where several definitions are valid, show how the applicable context is selected.

RESPONSE _____ EVIDENCE OR GAP _____

4 **Does the knowledge path preserve source controls?**

Can an answer cite an applicable, current source the caller may see? Demonstrate permission revocation, replacement, deletion and stale-index behavior. If retrieval sits outside the workflow, record that scope and the control owner's agreement.

RESPONSE _____ EVIDENCE OR GAP _____

5 **Is identity enforced through the complete path?**

Can the team demonstrate the caller and agent identities at the actual enforcement points? Test an unauthorized user, a revoked entitlement and the relevant tenant boundaries. Include APIs, databases, retrieval and caches that the workflow uses.

RESPONSE _____ EVIDENCE OR GAP _____

Readiness questions six to ten.

6 **Is the agent limited to approved capabilities?**

Show the tool inventory, typed interfaces, owners and allowed side effects. Demonstrate that a read request cannot become an unauthorized action or bypass the intended scope.

RESPONSE _____ EVIDENCE OR GAP _____

7 **Does evaluation block an unsafe release?**

Show the versioned test pack, thresholds, expected refusals and injection tests. Demonstrate that failures block release and that material changes trigger re-evaluation.

RESPONSE _____ EVIDENCE OR GAP _____

8 **Can one request be reconstructed?**

Pick an answer and trace its caller, authority, sources, component versions, policy result and outcome. Confirm the right reviewers can reach the evidence, and that it's kept without unnecessary sensitive content.

RESPONSE _____ EVIDENCE OR GAP _____

9 **Can the service fail safely and recover?**

Show service and freshness objectives, operating limits, support, rollback and tested recovery. Demonstrate source, policy and tool failures. For writes, show duplicate protection and remediation.

RESPONSE _____ EVIDENCE OR GAP _____

10 **Can the internal team operate it economically?**

Show cost per completed useful task, adoption, business results and who owns recurring maintenance. Have the internal team demonstrate routine support and change review. Confirm the basis for expanding the scope.

RESPONSE _____ EVIDENCE OR GAP _____

Interpret the answers without hiding a critical gap

YES _____ PARTIAL _____ NO _____ UNKNOWN _____

Every gap needs an owner and an action. A critical failure in access, isolation, unauthorized action, integrity or evidence blocks the affected release, whatever the other answers say. An applicability exclusion needs a documented scope boundary and control-owner acceptance. Don't count it as demonstrated evidence.

All Yes answers justify an evidence review. They don't justify automatic approval. Choose the most consequential gap and agree the test that closes it. Repeat whenever the workflow, users, sources or authority change.

First capability worksheet.

Use this page to turn the guide into a bounded proposal. Complete it with the process owner and the delivery team before committing to implementation. Link detailed evidence separately, so the decision stays easy to inspect.

Define the work and the result

Workflow and intended users

Business process owner

Current bottleneck and baseline

Expected outcome and measurement period

Human decisions retained

Bound the capability

Approved data products and source owners

Definitions that need a business decision

Knowledge corpus and content owner

Allowed tools and actions

Users and data explicitly outside scope

Agree on the release evidence

Most consequential diagnostic gap

Test that would close it

Risk and control owners

Service, freshness and cost limits

Internal operating owner and transfer demonstration

Record the investment decision

Decision Proceed / Reshape / Stop

Reason and remaining conditions

Next evidence review date

Bring the workflow that should already be in production.

The next conversation should start with a specific piece of work: the pilot that can't pass its control review, the recurring reconciliation that eats expert time, or the information task that should be faster but stays hard to trust.

Bring the completed worksheet, the architecture you inherited and one representative request. Include the business owner and someone who understands the data and access constraints. The working session can then identify the assumption that needs testing, and the smallest useful scope that would test it.

What RJL brings to the work

RJL Solutions connects strategy, architecture, engineering, governance and operation through Frame, Prove, Build, and Run & Transfer. The same business case should govern the build and the outcome review. Internal capability develops throughout delivery, so that transfer has a demonstrable end condition.

Each phase is separately scoped and priced in a written proposal before you commit. Request a free sixty-minute working session at **rjlsol.com/contact** or write to **rwhite@rjlsol.com**.

Richard White
RJL Solutions · Technology, data and AI. Strategy through production.

References and further reading

Architecture foundation

The Four-Part System (RJL Solutions, 2026) develops this architecture in vendor-neutral form: governed meaning, data-product contracts, controlled interfaces, production evidence and iterative release. Validate any implementation example for the selected environment before adoption.

RJL approach and scope

The companion papers are at rjlsol.com/whitepapers, including *The 90-Day First Governed Capability* (the weekly plan), *Agent Onboarding*, *AI Governance Evidence* (US bank and insurance expectations, NIST, and UK and EU rules) and the 2026 practitioner series on the operating model, data products, platform and contextual governance.

Risk management reference

The NIST AI Risk Management Framework 1.0 (January 2023, under revision in 2026) and its Generative Artificial Intelligence Profile (July 2024) provide voluntary risk-management guidance. For US banks, SR 26-2 (April 2026) replaced SR 11-7 and SR 21-8; it is nonbinding and most relevant above $30 billion in assets, and places generative and agentic AI outside its scope; *AI Governance Evidence* maps this guide's evidence to that and other regimes. Use the relevant guidance alongside your own obligations and control requirements. This guide supplies an operating pattern rather than a regulatory certification. Links updated 15 October 2026.

Scope of the examples

The claims workflow and the planning sequence are illustrative. Outcomes, timings, limits and release criteria have to be agreed and demonstrated for the actual workflow.

ABOUT

Richard White

I was global firm chief data officer at McKinsey & Company, owning data, AI and platforms end to end, and before that chief data officer at New York Life, where I built the chief data office and reported data risk to risk committees at multiple levels. Earlier I led data at the Federal Reserve Bank of New York and enterprise data governance at Citigroup. I founded RJL Solutions in 2024. My book, *The Day After the Demo*, is free at rjlsol.com/book.

RJL Solutions exists because the gap I kept being asked to close was never a strategy gap. The strategy was usually fine. What was missing was the part where somebody stays until it runs.

RJL SOLUTIONS

WEB rjlsol.com

EMAIL rwhite@rjlsol.com

ENGAGEMENTS Scoped and priced in a written proposal; first session free

This guide is a reference architecture and target-state operating model. It is not a product commitment, and it describes no client engagement or client result. Any implementation must validate platform maturity, regional availability, licensing, regulatory obligations, performance and operational ownership before production adoption.