RJL SOLUTIONS

TECHNICAL GUIDE / GOVERNED AGENTS

Agent Onboarding

Treating an agent like a data product: the contracts and evidence a governed enterprise capability needs.

CONTRACTS · EVALUATION · CONTINUING OWNERSHIP

Richard White
RJL Solutions · rjlsol.com

OCTOBER 2026

01 / THE PREMISE

Onboard the capability, not the connection.

An enterprise agent should enter production as an owned, versioned capability with an explicit contract. The contract says what work it performs, which information and tools it may use, how it's evaluated and who maintains it. Connecting a model to a tool creates a callable path. Onboarding establishes when people may rely on it.

This guide is for architects, engineering leads, data product owners, and the business and control owners who approve a release. It extends data product discipline to the agent's interface, dependencies, authority and operation. Named people stay accountable for the agent. It can't approve its own permissions or certify its own output.

Separate the agent contract from the tool contracts

The agent contract defines the business task, the intended users, the allowed outcomes, the exclusions and the service commitments. Each tool contract defines a specific operation, input and output schemas, authorization, side effects, failure behavior and version compatibility. A certified data product behind a certified tool doesn't make every agent that calls it production-ready.

Approve the complete composition, for a stated scope. That covers the model, prompts, routing, retrieval, memory where it applies, tools, source products, policy and user interface. Change any of them and approved behavior can change.

Use explicit lifecycle states

STATE	WHAT THE STATE REQUIRES
Proposed *(intake)*	Business owner, useful task, exclusions, and dependency assessment
`sandbox`	Feasible behavior within approved experimental data limits
`managed_sandbox`	Real data controls, named owners, and tested request paths
`certified`	Accepted evaluation, enforced authority, release evidence and support, with certified dependencies
`degraded`	Restrictions after a recorded breach; returns only when the cause is fixed and retested
`deprecated`	Replacement and migration deadline given; agreed service continues
`retired`	Terminal: consumers migrated; access, endpoints and retained data addressed

Proposed is intake. The other six are the Data Product Certification Standard states, where `certified` requires version 1.0.0 or later, so the 0.1.0 sample is pre-certification. This is not an external certification scheme. Record approved users, tools, versions and conditions with each decision; a catalog entry must not widen authority. The sample contracts and claims workflow are illustrative.

02 / THE CONTRACT

Publish a versioned agent product contract.

Store the contract beside the implementation, and promote both through review and release. Before consumers adopt the agent, they need to know the approved task, interface, dependencies, restrictions, support route and compatibility rules.

AGENT-PRODUCT.YAML — ILLUSTRATIVE STRUCTURE

```yaml
schema_version: rjl.agent-product.v1
id: claims-context-assistant
version: 0.1.0
lifecycle: managed_sandbox

owners:
  business: claims-process-owner
  product: claims-assistance-team
  service: claims-platform-operations
  control: claims-risk-and-security
scope:
  task: assemble_claim_context
  users: authorized_claims_examiners
  excluded: [adjudication, payments, external_contact]
  interface: contracts/claim-context-response-v1.json

dependencies:
  data: [claims.claim-status@0.1.0]
  meaning: [claims.claim-relationships@0.2.0]
  knowledge: [claims.procedures@0.3.0]
  tools: [get_claim_context@0.1.0, find_claim_procedure@0.1.0]

identity:
  mode: delegated_user
  workload_ref: identity/claims-assistant-svc-v1
policy_ref: policies/claims-read-v4
runtime_ref: releases/claims-runtime-v1
memory_policy_ref: policies/session-only-v1

limits:
  max_tool_calls: 6
  deadline_seconds: 30
  on_limit: escalate
evaluation_ref: evaluations/claims-context-v1
service_ref: operations/claims-service-v1
evidence_ref: evidence/claims-release-v1
approval_ref: approvals/managed-sandbox-001.json@a41e7c2
```

Every `_ref` stands for a versioned artifact a reviewer must be able to open. `identity.mode` says whether authority is delegated from the caller or is the service's own, and `workload_ref` names the agent's own workload identity in either mode; with delegation, effective authority is the intersection of user and workload. `on_limit` says what happens at a limit: stop or escalate, with a reason. `owners` names who answers for the outcome, the product, the service and the controls; `dependencies` lists the versioned products, meaning, knowledge and tools the agent may use, and nothing else. The limits are illustrative.

Make the declarations testable.

A release validator should check the manifest against its JSON Schema (2020-12, `additionalProperties: false`), resolve the references, and compare the deployed configuration with the approved artifacts. The runtime reference should identify the model configuration, prompts, routing, tool schemas and the relevant retrieval configuration. Put credentials in the approved secret or identity system, outside the manifest.

THE POINT OF THE FILE The manifest supplies declarations. Enforcement code, tests and operational evidence establish whether those declarations are true. A complete file cannot compensate for an unrestricted tool or an unowned dependency.

Certify the information the agent consumes

Map each supported user question to the information needed to answer it. Identify the authoritative source, the approved meaning, the applicable knowledge and the permitted interface. This mapping exposes dependencies that a fluent demonstration can hide.

DEPENDENCY	COMMITMENT AND EVIDENCE THE AGENT NEEDS
Data product	Owner, grain, identifiers, quality, freshness, lineage, and supported interface
Business meaning	Approved entities, relationships, calculations, exceptions, and known answers
Knowledge corpus	Applicable versions, source provenance, permissions, and update or deletion behavior
Tool interface	Typed inputs and outputs, allowed operations, compatibility, and failure behavior

For claims context, tell the insured person apart from the policyholder or the employer. Use approved relationship rules, and report unresolved matches. Current status should come from the designated operational interface, or from a product whose freshness meets the task's requirement. A benchmark question should use its approved metric definition and reporting period.

Preserve controls through retrieval

Apply entitlement and applicability restrictions before restricted passages enter model context. Carry source identifiers, effective dates and product or jurisdiction metadata into the retrieval path. Citations should let the authorized user inspect the evidence behind the answer.

Push a replacement procedure, a deleted document and a revoked permission through the indexes and the relevant caches, and test the result. Define how long updates may take, and what behavior is allowed in the meantime. A corpus known to be stale should say so, or restrict the affected task as the contract requires.

Bound memory as a separate dependency.

Separate request or session context from persistent memory. Declare the allowed persistence, the readers, the writers, the retention, and how deletion and revoked authority affect reuse. Validate model-generated statements before accepting them as authoritative business facts.

The worked example in this guide uses session-only context and no persistent agent memory. If persistence is added, reassess its isolation, provenance, write controls and evaluation. OWASP treats memory poisoning and cross-session influence as agent security concerns.[1]

Record the state or as-of time of changing source data where relevant, and maintain compatibility rules for sources that update independently of the agent release.

Enforce identity and bound execution

For delegated work, three things bound effective authority at once: the caller's entitlement, the approved scope of the agent or tool, and the request's permitted purpose and context. A background service needs its own explicitly approved workload authority. It shouldn't inherit the permissions of a user who's no longer taking part.

Enforce the decision outside the model

The execution component has to validate identity, target, operation and policy before a tool does the work. Restrict network destinations and tool availability to the approved capability.

> **The model may propose an operation. It cannot grant the authority to execute it.**

Transport authentication is one part of the path. For Model Context Protocol (MCP) integrations, follow its audience validation and the prohibition on token passthrough.[2] Never forward the caller's token. Each hop validates the token's audience and obtains a narrowly scoped token for the next, for example through OAuth 2.0 Token Exchange.

04 / THE REQUEST PATH

Define the complete request path, and certify actions separately.

Seven steps. Each has to hold under retries, asynchronous work, pooled connections and caches. A gateway check doesn't show that a downstream query or a cached response respects the same boundary.

01 Authenticate the caller and identify the agent release.

02 Resolve the task and the applicable business context, and clarify ambiguity.

03 Select only a registered tool allowed by the approved scope.

04 Validate typed arguments, and authorize the exact operation and target.

05 Enforce source, row, field, document and tenant restrictions where applicable.

06 Validate the returned result, assemble a bounded answer, and preserve citations.

07 Record the policy outcome, the tool result and the response under one trace identifier.

Define safe behavior for the case where identity or a policy decision cannot be established at all.

Certify consequential actions separately

Moving from reading to writing is a scope change. Define the business command, its validation, approval, duplicate protection, durable outcome and remediation. Where approval is required, bind it to the actor and the exact action parameters, and validate it at execution. A model-supplied confirmation flag is insufficient evidence.[1]

Prevent unbounded execution with tool-call, duration, concurrency, retry, retrieval and cost limits that suit the workflow. At a limit, stop or escalate with an explicit reason. A fallback has to preserve authority. It must never present an incomplete task as complete.

Bound what the agent may do, and make every write safe to repeat.

The contract declares how much the agent may do. Two questions set that bound: what would an error cost, and how much human judgment does the task need? Record the quadrant and its control in the agent product contract.

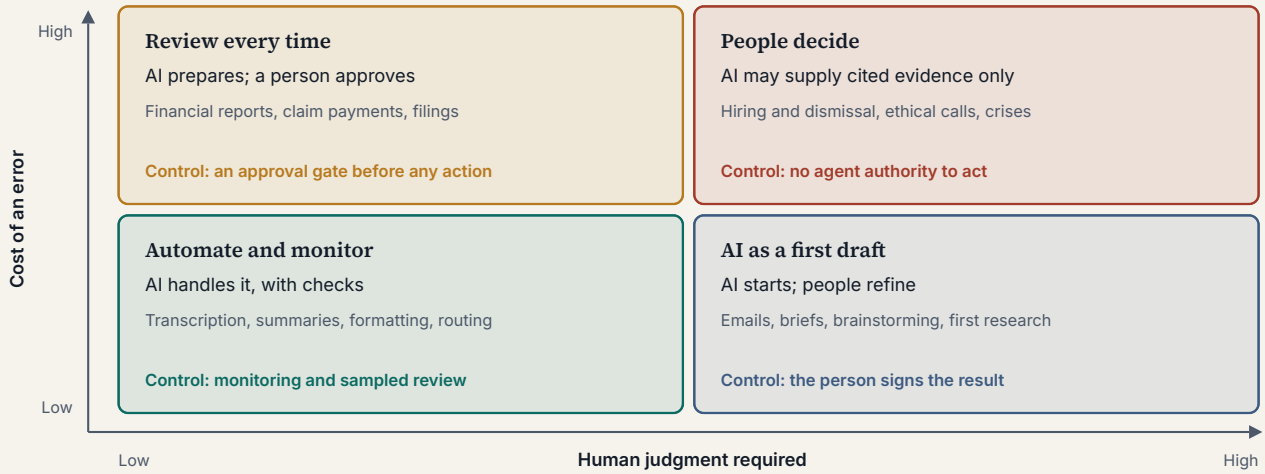
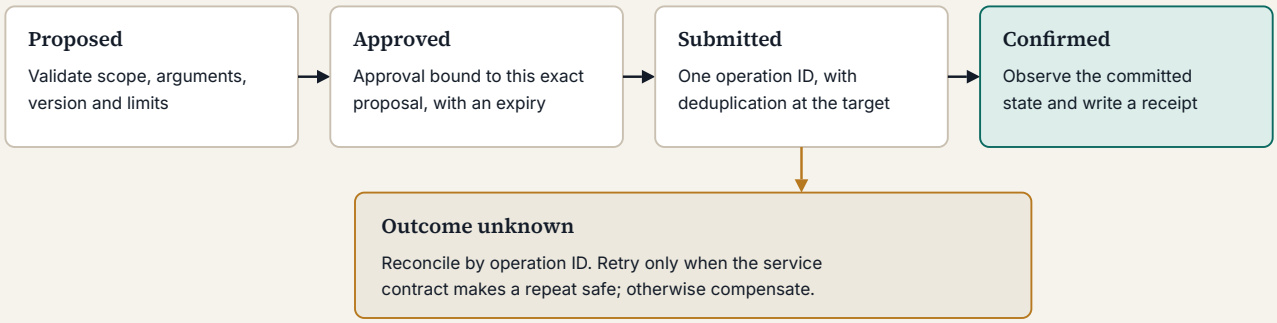


Exhibit: The cost of an error and the judgment required decide how much AI may do.

Give every consequential write an action contract

Bind the approval to the exact proposal and give it an expiry, so a changed argument needs a new approval. Give each intended write one operation ID, and have the target deduplicate on it. When a response is lost, treat the outcome as unknown: reconcile by operation ID before any retry, and retry only where the service contract makes a repeat safe.[3] Decide in advance which actions can be rolled back and which need a compensating action, such as a reversal entry.



A timeout reports missing confirmation. It does not prove that the business action failed.

Exhibit: An action needs a known state before it can be retried.

THE LOST-RESPONSE TEST

Drop the response to a submitted write during testing. The release passes only if the agent reconciles the outcome by operation ID, and neither repeats the action nor reports it as failed.

05 / EVALUATION

Evaluate useful results and control behavior.

Define acceptance criteria before evaluating a release. Separate deterministic requirements, such as authorization and source selection, from probabilistic answer quality. An aggregate accuracy score can't excuse an unresolved failure in disclosure, isolation, unauthorized action, integrity or evidence.

Build a versioned evaluation pack

For each scenario, record the user and authority, input, source state, expected route, allowed tools and observable result. For interpretive answers, define an explicit rubric, and use reviewers with enough context to judge correctness and support.

SCENARIO FAMILY	EVIDENCE TO COLLECT
Representative tasks	Useful completed work, correct meaning, sources, and required citations
Ambiguity or conflicting evidence	Clarification, disclosed uncertainty, or escalation to the owner
Revoked or unauthorized access	Denial across tools, sources, retrieval, and the relevant caches
Hostile prompts or documents	Tool boundaries preserved and prohibited actions blocked
Failed or stale dependencies	The contract's restriction or recovery behavior
Repeated or expensive work	Retry and execution limits, attribution, and completion state

Pair a permitted request with a near-identical prohibited one. Test multi-step paths where a valid request tries to widen its scope. Retest failure cases after fixes, and keep them as regression scenarios.

Interpret results within the tested scope

Where model behavior varies, run enough repetitions to understand the variation that matters, and record the settings and sample sizes. Tune thresholds to the business task and the consequences of error. Don't pick a universal pass percentage. Don't treat a small sample as a guarantee either.

Measure answer usefulness after human verification and corrections. Record unsupported assertions, incomplete work, and refusals that blocked a permitted task. A safe refusal can be correct behavior. Too many refusals can still make the capability unusable.

WHEN A CONTROL FAILS	Block the affected release until it is resolved, or until the scope is redesigned and re-evaluated. The release decision must identify exactly which configuration and scope the evidence supports.

06 / RELEASE

Promote with inspectable release evidence.

Promotion changes the obligations of the service. A managed sandbox can investigate a bounded task under approved controls. A production release has to demonstrate its consumer commitments and have an owner ready to support them.

Keep a release record the team can inspect

ARTIFACT	WHAT A REVIEWER SHOULD BE ABLE TO ESTABLISH
Agent and tool contracts	Approved scope, exclusions, schemas, owners, and compatibility
Configuration and dependencies	The named version combination: runtime artifacts, model, prompts, policy, sources and tools
Evaluation report	Scenario results, settings, unresolved gaps, and accepted thresholds
Access and execution evidence	Actual enforcement points, and permitted and denied request paths
Operations evidence	Monitoring, limits, incident route, rollback, and recovery exercise
Approval record	Decision holders, conditions, accepted scope, and review triggers

Reference controlled artifacts. Don’t copy sensitive case data into the release record. Use integrity controls suited to the enterprise to tie approvals and evidence to the release. A digest can identify an artifact. It can’t prove the artifact behaves correctly.

Make the deployment gate enforce the decision

Validate required fields and references, run the applicable evaluation and block promotion when critical conditions fail. Confirm the deployment matches the named, reviewed combination, with the required limits and monitoring turned on. Keep deployment access separate from the authority to accept the business outcome or a control exception.

Judge the release on business, control and operational evidence together. A favorable average can’t override a failed critical control. A review should settle whether to release the stated scope, reshape it or stop. Conditions need owners, closure evidence and enforcement wherever they affect runtime behavior. A narrower release needs its own coherent contract and evidence. Dropping a failed test from the report doesn’t resolve the failure.

Reconstruct a request after release

Link the caller and workload identities, the release identifier, the selected definition, source provenance and as-of state, policy decisions, tool calls, response and observed outcome. Define how reviewers retrieve the trace, and how long the relevant evidence is kept. Reconstruction supports investigation and explanation. It doesn’t promise byte-for-byte reproduction of a probabilistic answer, and it doesn’t mean keeping every sensitive input forever.

07 / OPERATION

Operate through change and failure.

An agent's certification is a continuing commitment for a scope. The service owner needs to see its outcomes and dependencies, needs the authority to restrict it and needs a funded process for keeping the commitments after the original delivery team leaves.

Measure the service people actually consume

Declare service objectives for useful completion, response time, availability and, where relevant, information freshness. Define the denominator, the measurement window, the exclusions, the alert thresholds and the response to a breach. Include review effort and exceptions in the business result.

Track cost per completed useful task, covering model, retrieval, integration, review, support and maintenance. Attribute it to the agent and the calling identities, without secrets or unnecessary personal information in telemetry. Monitor rejected and abandoned work too.

Review a scorecard with the owner: test accuracy, intervention and override rates, escalations, drift over 30 to 60 days and cost per task. Interventions should fall as the agent earns trust. A rate near zero can mean reviewers have stopped reviewing.

Respond to degraded dependencies deliberately

CONDITION	BEHAVIOR TO AGREE ON AND TEST
Entitlement cannot be verified	Deny the affected request rather than assume access
Required source is stale or unavailable	Restrict, clarify, or route to an approved alternative
A tool reaches its limit	Stop, or queue eligible work with a visible reason
A changed dependency breaks compatibility	Restrict the affected route and restore an approved configuration
An action outcome is uncertain	Reconcile its durable state before retry or remediation

Name who can disable a tool, narrow a user population, roll back a release or suspend the service. Preserve incident evidence and tell users what the effect is. For write workflows, a software rollback may not reverse a completed transaction. Remediation needs its own authorized process.

Control changes and retirement

Use a change assessment to find affected contracts, consumers, evaluation and operational evidence. Keep independently updated knowledge sources and policies inside that process. Recheck permissions when caching or reusing prior output, and define when cached content gets invalidated.

Retirement should remove discovery and execution access, disable unused identities and endpoints, migrate consumers and deal with retained memory and evidence under the approved retention policy. Mark the catalog entry, so nobody presents a retired version as a supported capability.

08 / OWNERSHIP

Assign decisions, and prove internal transfer.

The operating model needs people who can change the business process, maintain dependencies, enforce controls and support the service. A product owner coordinates this work. They can't substitute for the domain or control authority a decision needs.

RESPONSIBILITY	DECISION AND EVIDENCE THE ROLE OWNS
Business process owner	Outcome measure, workflow changes, adoption, and acceptance
Agent product owner	Scope, lifecycle, consumers, backlog, and release coordination
Domain and source owners	Meaning, quality, freshness, knowledge applicability, and change notices
Security and control owners	Permitted use, enforcement conditions, escalation, and control approval
Service owner	Support, limits, recovery, recurring cost, and service restrictions

A small team can combine roles while keeping the responsibilities explicit. Record who may accept residual risk and who can stop a release under the organization's process. Set a route for disputed definitions, unreachable sources and unresolved control conditions.

Divide shared services from domain authority

Shared platform teams should maintain reusable identity, deployment, evaluation tooling, monitoring and catalog infrastructure. Domain teams keep the authority to approve meaning and change the business workflow. Agree the interface between them, including who responds when a dependency or a platform service breaches its commitment.

Fund ongoing ownership alongside delivery. Source upkeep, evaluation, incident response and consumer support all need capacity after the initial build. Give internal engineers access to the implementation, the artifacts and the operating tools early enough to learn through delivery.

Require a handover demonstration

Have the receiving team onboard a permitted user, revoke access, investigate a request, assess a source change and carry out a rollback or restriction in the agreed environment. They should explain the expected outcome and produce the evidence, without leaning on the original implementer to fill in missing steps.

TRANSFER IS COMPLETE WHEN	The team can perform its responsibilities with the required authority and support. Record remaining gaps with named owners and acceptance conditions. A runbook, a repository, or a training attendance record alone does not establish operating readiness.

09 / WORKED EXAMPLE

Work through one onboarding decision.

Take a claims-context assistant that assembles case facts and finds the applicable procedure. Authorized examiners stay responsible for the case decision. Adjudication, payment, record updates and external communication are excluded. This example illustrates a review. It reports no deployment and no test results.

Define the release the evidence must support

The assistant uses `claims.claim-status@0.1.0`, `claims.claim-relationships@0.2.0` and `claims.procedures@0.3.0` through two approved read tools, and retains session-only context. The output contract separates source facts, supporting references, interpretation, unresolved matches and incomplete work.

REVIEW QUESTION	EVIDENCE REQUIRED FOR THIS PROPOSED SCOPE
Is the right case and person selected?	Known cases and ambiguous relationships exercise the approved matching rules
Is the applicable procedure returned?	Product, date and permissions select an authorized source version
Does revoked authority take effect?	Source, retrieval, tool and cache tests demonstrate denial
Can content induce a prohibited action?	Hostile requests and passages leave write and external tools unavailable
Can the internal team support it?	Trace review, change assessment, restriction and recovery

Record the decision and the boundary

An entitlement failure blocks the affected scope. Fix a procedure source that misses its update commitment, or redesign and retest the task. Before deployment, record the permitted users, versions, conditions and decision holders.

> **Approval of this read capability does not authorize a payment tool.**

Use the **10-Question Diagnostic** for the wider session and the **Blueprint** for the architecture. Bring one proposed agent, its dependency map and its open decisions to **rjlsol.com/contact**.

[1] OWASP AI Agent Security Cheat Sheet. Tool authorization, memory security, adversarial validation.

[2] MCP Security Best Practices, 25 November 2025. Token audience validation and passthrough.

[3] Amazon Builders' Library, "Making retries safe with idempotent APIs." Client request identifiers and the handling of unknown outcomes.

Proposed operating guidance. Enterprise policy and the named decision holders determine approval.

PRODUCT CONSTRUCTION / CONSUMER READINESS

Onboard the agent as a real product consumer.

An invoice assistant needs a supported contract for every analytical, live and knowledge dependency.

DEPENDENCY	REQUIRED BOUNDARY TEST
Dated receivables product	Return the definition and cutoff with the balance. Deny out-of-scope entity/tenant data; reject an unsupported version.
Live invoice interface	Use current state for time-sensitive requests. A dated analytical snapshot cannot silently substitute for a current response.
Policy knowledge	Retrieve only permitted versions and passages; test revoked rights, source deletion, stale indexes and cached content.
Proposed action	Use typed inputs and current authority. Bind required approval to the exact payload and state; deny changed or expired proposals.
Operation	Record source and policy references, evaluated behavior and result. Reconcile an uncertain external write before any retry.

Evaluate the synthetic T0 balance of 500, the later balance of 300, an ambiguous currency, multiplied join totals and a request to change a live hold. Expected behavior includes answer, clarification, denial or escalation; a fluent answer is not always the correct outcome.

A new source contract, semantic mapping, permission model or materially changed product condition can invalidate the earlier agent evaluation. Use the dependency manifest to determine what must be retested. See Data Products in Practice for the synthetic inputs and calculation scope.

Design references reviewed 29 October 2026: Databricks: medallion architecture; Designing data products; Google SRE: service objectives. The worked records, controls and targets are illustrative RJL recommendations.

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 connects strategy, architecture, engineering, governance and operation through Frame, Prove, Build, and Run & Transfer. Each phase is separately scoped and priced in a written proposal before you commit. The first sixty-minute working session is free.

GUIDE **The Connected Enterprise Data & AI Blueprint**
Where this contract sits in the wider architecture, in 24 pages.

WORKSHEET **The 10-Question AI Production-Readiness Diagnostic**
Four pages to complete with your business owner, technical lead and control owner.

PERSPECTIVE **Why Enterprise AI Stalls After the Demo**
The executive case for everything described here, in ten pages.

RJL SOLUTIONS

WEB rjlsol.com

EMAIL rwhite@rjlsol.com

This guide describes no client engagement and no client result. The claims workflow, the sample contracts and the lifecycle states are illustrative. They aren't an industry schema. Before production adoption, any implementation has to validate platform maturity, regional availability, licensing, regulatory obligations, performance and operational ownership.