RJL SOLUTIONS

REFERENCE ARCHITECTURE / IMPLEMENTATION

The Reference Implementation

AI, data and security, built and operated as one system, with two use cases traced end to end.

LAYERS · LABELS AND IDENTITY · ORCHESTRATION · A BANK AND AN INSURER

Richard White
RJL Solutions · rjlsol.com

OCTOBER 2026

01 / THE PREMISE

One system, not three programs.

Most enterprises fund AI, data and security as separate programs, each with its own platform, team and roadmap. Production AI fails at the seams between them. An agent reaches data its user may not see. Two teams answer the same question with different numbers. Nobody can say who approved a change. This paper shows one way to build and run the three as a single system.

I wrote it for enterprise architects, chief data officers, chief information security officers and the heads of data, AI and platform engineering at US banks and insurers, and at any firm that has to show its controls to an auditor, a customer or an investor. It's the implementation companion to *The Four-Part System*, which describes the same architecture without naming products.

I designed this architecture and refined it through hands-on implementation in large regulated enterprises. It names real products as one example combination, checked against vendor documentation in October 2026. Each product is an example of its pattern. Any substitute has to keep the same contracts and controls.

SECTIONS	WHAT THEY COVER
02 and 03	The full picture, and the layers with example products
04 to 07	Data platform, meaning and knowledge, interfaces, and AI
08 to 11	Security: labels and tags, agent identity and supervision, the security operations center, and one owner per control
12 and 13	Orchestration and the operating model that hold it together
14 to 17	Two use cases traced end to end: a bank and an insurer
18 to 20	The implementation path, a readiness checklist, and sources

DISCLOSURE The author has worked closely with several vendors named here, including Cyera, ServiceNow, Snowflake, Databricks and Microsoft, as a customer and practitioner. No vendor reviewed or paid for this paper. Products are named as examples of the patterns, not endorsements.

02 / THE FULL PICTURE

Design AI, data and security together.

Governed production AI depends on three capabilities, and on the three places where each pair has to be designed together. Orchestration and the operating model hold them together day to day: the process, control plane, release gates and accountable owners.

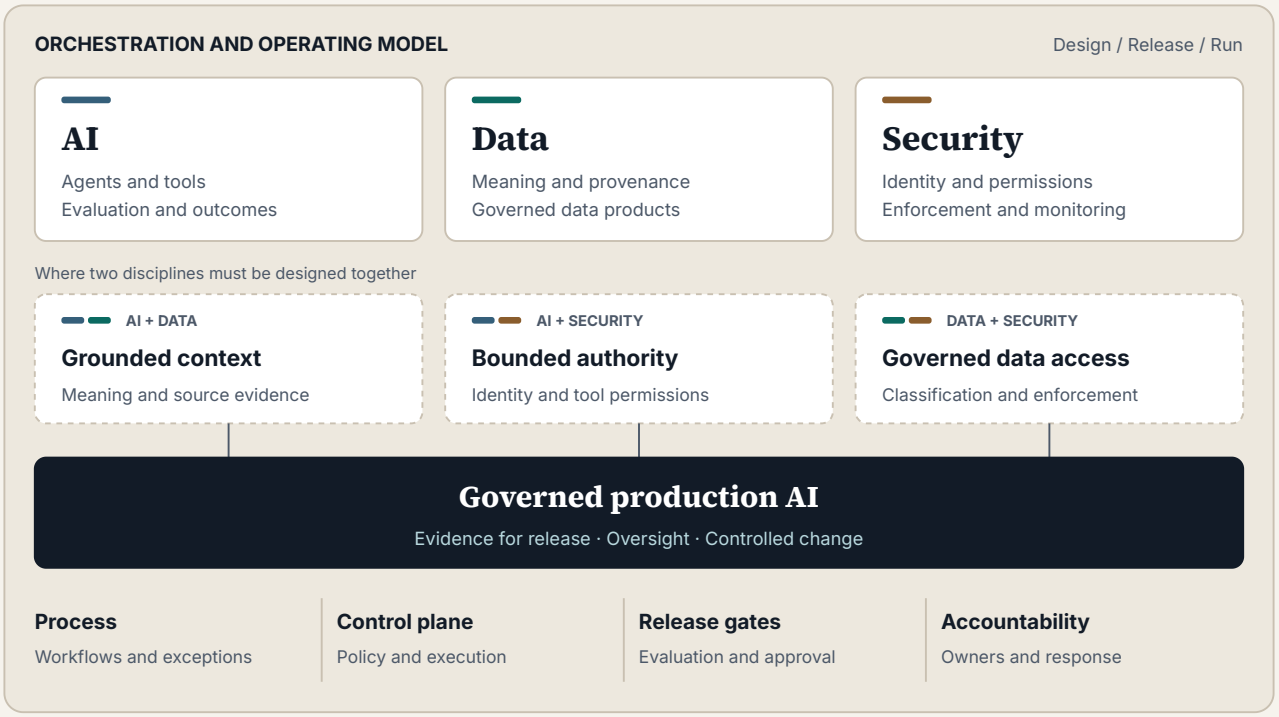


Exhibit: Design AI, data and security together. Three capabilities, one accountable operating model. A conceptual model, not a deployment topology.

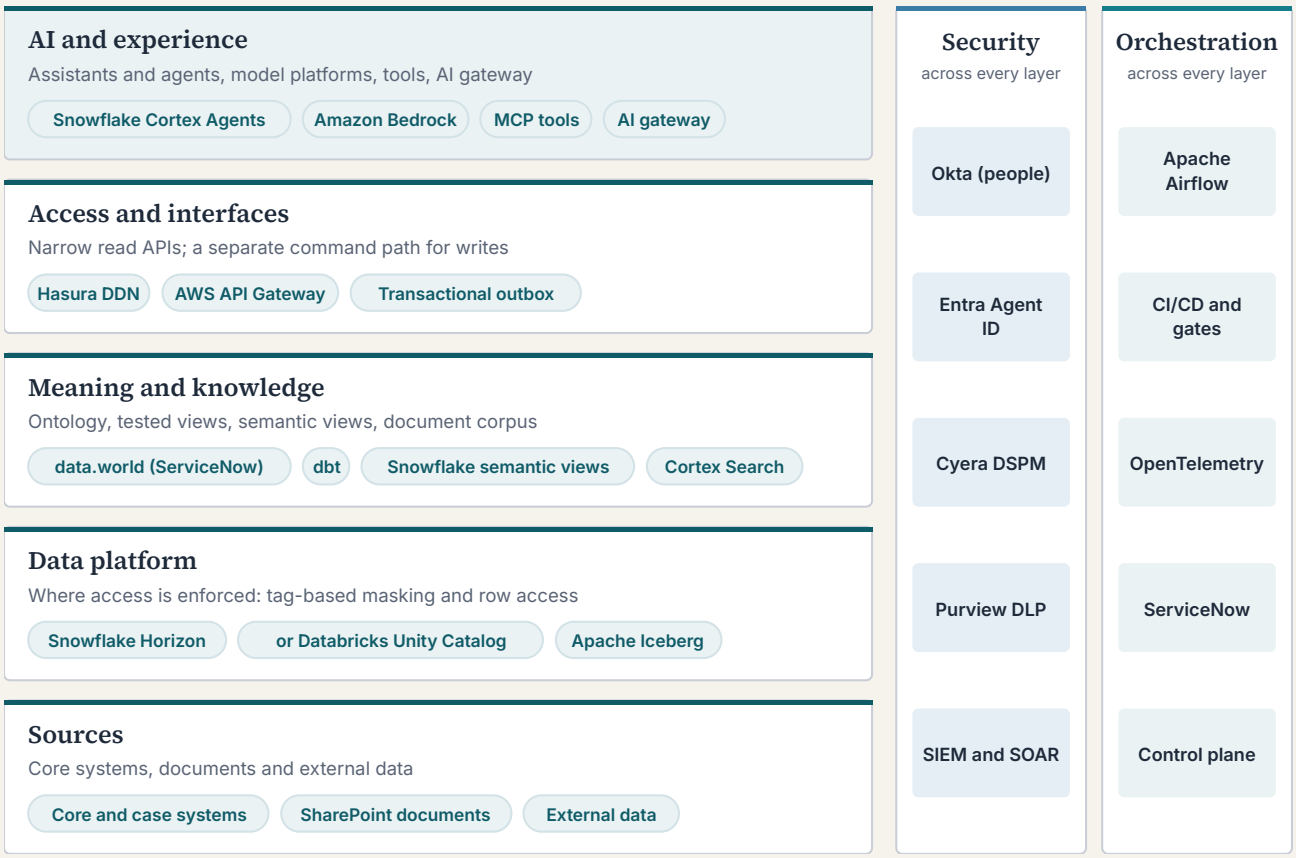
WHERE TWO MEET	WHAT IT REQUIRES
Grounded context (AI and data)	A semantic layer the business approved, certified data products, and retrieval that applies each user's permissions before the model sees anything
Bounded authority (AI and security)	An identity for every agent, authority bounded by the user and the tool, and defense against instructions hidden in content
Governed data access (data and security)	One governed vocabulary, propagated to approved serving paths; binding coverage, drift and revocation lag tested
Governed production AI	All three at once, released on evidence, overseen and changed under one control plane with named owners

In the terms of *The Four-Part System*, data is the brain, security is the immune system, and orchestration is the nervous system and the body. The AI capability is what the business uses on top of all four.

03 / THE LAYERS

The architecture, with one example combination of products.

Five layers carry a request from the user to the source. Security and orchestration run through every layer. They don't sit on top of it.



The diagram shows Snowflake as the data platform. Databricks with Unity Catalog fills the same role, with governed tags and attribute-based row filters and column masks generally available since May 2026. The contracts between layers matter more than the products. A team can replace any box, provided the replacement keeps the same interface, enforcement and evidence.

THE RULE THAT HOLDS IT TOGETHER

Every layer has one job. An enterprise metadata catalog doesn't enforce anything. A semantic model isn't an access policy. And an API gateway isn't a semantic layer.

04 / DATA PLATFORM

Enforce access where the data lives.

Every request ends up in the physical layer, whether it arrives through an API, a semantic view or an agent. That's where access gets enforced, so a rule written once holds whichever path a request took.

LAYER	ROLE	CONTROL
Bronze	Raw data as received from sources	Classified on arrival; restricted to engineering
Silver	Cleaned, conformed and linked records	Tags applied and propagated; quality checks
Gold	Published analytical products, each a maintained promise to its consumers	Contracts, service levels and tag-based policies

Tags drive the policies

Classification tags support masking and row access only where a correctly bound policy enforces them. Verify the effective caller and policy through every serving path. In Snowflake, tag-based masking policies are generally available and tags can propagate to dependent objects; tag-based row access policies entered public preview in July 2026, so until they are generally available, attach row access policies directly. Databricks Unity Catalog offers governed tags and attribute-based policies. Snowflake's tag-based row access preview needs matching policy arguments and column names or signature aliases: without a match, the policy is not enforced. Test binding coverage and denied queries [Snowflake documentation].

Treat federated sources with care

Data queried in place from another system sits outside the platform's enforcement boundary. It doesn't inherit the platform's policy by implication. Either ingest it under the same controls, or enforce an equivalent policy at its own connector and prove the two stay aligned. The same goes for open tables read by an outside engine.

EXECUTION BOUNDARY — The meaning layers above give a query its context. They never decide what a caller may see; the data platform does, every time.

05 / MEANING AND KNOWLEDGE

Give AI the meaning the business agreed.

An assistant can be only as consistent as the definitions it uses. The meaning layers turn raw tables into the entities, relationships and metrics the business has approved, and make documents retrievable under the same rules.

COMPONENT	EXAMPLE	JOB
Ontology	data.world, now part of ServiceNow	The agreed entities and relationships, such as customer, account, policy, claim and party, in the enterprise catalog
Tested views	dbt, coordinated by Apache Airflow	Views generated from the ontology mappings and tested against known answers
Semantic views	Snowflake semantic views	Approved metrics and definitions, versioned and owned, that agents can query
Knowledge corpus	Cortex Search, or a governed vector index	Policies, procedures and documents, indexed with their classification and entitlement labels

Bootstrap the ontology, then let experts own it

A language model can draft entities and relationships from existing schemas and documents. Subject experts then correct and approve them. The approval makes the ontology authoritative. The draft doesn't.

Documents follow the same rules as tables

A contract in a document library needs the same tenant and sensitivity labels as a row in a table. A search index is a derived copy, and Cortex Search runs with its owner's rights. So sync source permissions into the index, and filter each query by the entitlements in the verified token before anything reaches the model. Test that retrieval denies whatever the source would deny.

MEANING IS CONTEXT, NOT AUTHORITY — Semantic and ontology layers help the system choose the right answer. Access is still decided at the data.

06 / INTERFACES

Expose narrow operations, not the warehouse.

Agents and applications reach data through narrow, versioned operations. Open query access isn't offered. Reads and writes take different paths.

CONTRACT	WHAT IT DEFINES
Data product	Meaning, owners, quality, freshness and service levels for one product
API	Operations, schemas, authorization and rate limits; in this combination, Hasura DDN over the platform, behind AWS API Gateway
Tool	One operation an agent may call, its side effects, and how it fails, published through the Model Context Protocol (MCP)
Agent	The business task, users, allowed outcomes, exclusions and service commitments

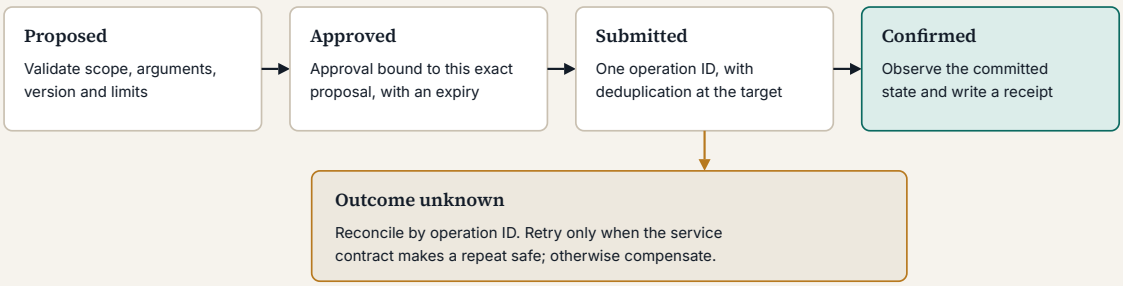
Keep four contracts. One supergraph would tie them together. Each can change on its own schedule, with its own owner and tests.

Keep API permissions in step with classification

When the API layer applies its own permissions, it has to stay in sync with the classification that drives the platform's policies. A classification change should reach both enforcement points together, with someone owning drift detection.

Writes take a separate, certified path

The first release reads. A write to a system of record goes through a separately certified command path with an action contract. That contract covers validation; approval where the risk requires it, bound to the exact proposal and given an expiry; one operation ID per intended write, deduplicated at the target; reconciliation of an unknown outcome before any retry; a defined rollback or compensating action; and a transactional outbox, so the business change and its record can't drift apart.



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.

TOOL SECURITY Follow the MCP security guidance: validate each token's audience, never pass a caller's token through, and obtain a narrowly scoped token for each downstream hop.

07 / AI AND AGENTS

Run agents as governed products.

The AI layer plans the task, calls tools and composes the answer. Cost, routing and release discipline live here too.

COMPONENT	EXAMPLE	JOB
Agent runtime	Snowflake Cortex Agents; Amazon Bedrock for external models	Plans and executes within the agent contract
Gateways	A model gateway for external models; an MCP gateway for tools	Apply policy, check labels, redact, log and limit cost; in-platform models rely on platform controls
Router	A versioned routing policy	Resolves the persona and intent of a request, and sends it to the right agent and data
Evaluation suite	Versioned test sets in the release pipeline	Quality, refusal and attack tests; a failure blocks release
Memory	Governed session and long-term stores	Each memory class has an owner, a classification and a retention rule

Make the router a governed policy

Routing decides which agent, tools and data a request reaches. Treat it like any other policy: versioned, reviewed, tested and recorded in the trace for every request.

Price every completed task

Track inference, platform, review and rework against completed useful tasks, with budgets, showback and limits per capability. If a capability's supervision costs more than it saves, redesign it or stop it.

RELEASE DISCIPLINE — Every change to a model, prompt, tool, route or policy passes the same evaluation gate as a code change.

Propagate labels; verify every approved path.

Agents act faster, and at greater scale, than human workflows. So systems have to enforce access limits; training and policy can't do it. Weak classification can affect both what an agent sees and what it's allowed to do, turning a data issue into an operational risk. Apply the constraint through identity, policy and labels that each approved enforcement point can interpret.

STEP	WHAT HAPPENS	EXAMPLE
Discover and classify	Sensitive data found and classified continuously across tables and documents	Cyera DSPM
Tag	One vocabulary: sensitivity, tenant, purpose and regulated categories such as customer financial data	Platform tags and catalog labels
Enforce	Tags drive masking and row access, retrieval filters, API permissions and data loss prevention	Snowflake Horizon or Unity Catalog; Microsoft Purview
Reconcile	Drift between the policy source and each enforcement point is detected and owned	Policy propagation checks
Evidence	What was allowed, masked or denied, and why	Audit store and trace

Where a direct integration exists, classifications get written into the platform's tags. Otherwise a governed tagging workflow applies them. Cyera's AI-focused modules add a view of which data each AI system can reach. A tenant label restricts answers only when the serving path enforces it, so test cross-tenant denial.

CLASSIFICATION IS NOT AUTHORIZATION A DSPM tool informs the controls that govern each surface. It does not replace them, and it should not become a second, competing control.

Register, credential, supervise and offboard every agent.

Use the employee lifecycle as a starting pattern. Give each agent a registered identity, an approved purpose, defined permissions, monitoring and a retirement process. Then impose tighter limits than you would on a person. An agent has no independent judgment, and the content it reads can steer it.

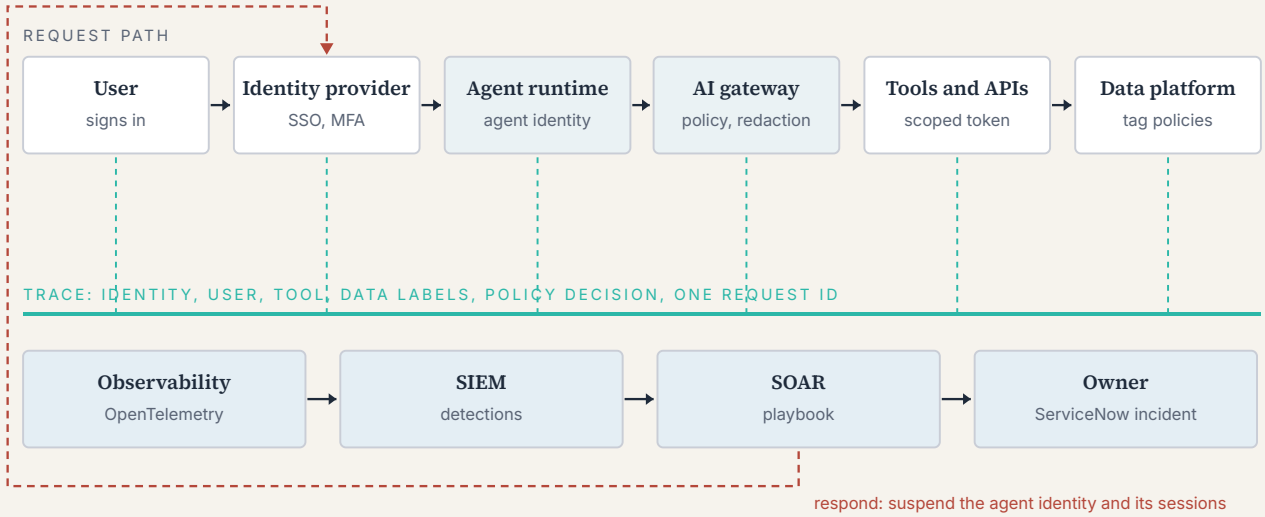
EMPLOYEE PRACTICE	FOR AN AGENT	EVIDENCE
Hiring	Registered with a named business owner, purpose and approved scope	Agent record and approval
Credentials	Its own identity, short-lived tokens, no shared or embedded keys	Identity record and token policy
Role	Role-based entitlements and a short tool list; acting for a person, only what both may do	Entitlement review and refusal tests
Manager	The business owner supervises; consequential actions need a person’s approval	Approval records
Monitoring	Every action logged with its identity and the data labels it touched	Traces and alerts
Access reviews	Roles, tools and data scope recertified on a schedule	Review sign-off
Offboarding	Identity, credentials, endpoints and memory removed at retirement	Retirement record

The identity mechanics

People sign in through the workforce identity provider, for example Okta, federated with Microsoft Entra. Agents receive their own identities, for example through Microsoft Entra Agent ID. Delegation issues a short-lived token bound to both the user and the agent, through OAuth 2.0 Token Exchange or the platform’s on-behalf-of flow (Entra uses the JWT bearer grant), with resource indicators so each token works only for its intended service. One switch disables the agent identity and its sessions. Issue short-lived tokens and use continuous access evaluation where the service supports it; otherwise an issued token stays valid until it expires.

Every agent action reaches the security operations center.

Design the request path and the telemetry path together. Each step emits a trace carrying the agent identity, the user, the tool, the data labels touched and the policy decision, under one request ID.



THE SOC WATCHES FOR	TYPICAL RESPONSE
Queries far outside the user's assigned work, or unusual volume or hours	Suspend the agent identity and open an incident for the owner
Access to data labeled restricted, or cross-tenant attempts	Deny, preserve evidence, alert the data owner
Tool calls outside the approved list, or attempts to widen authority	Block the call, quarantine the source, review the route
Instructions hidden in retrieved content	Flag the document, restrict the corpus, add the case to the attack tests

Classification findings reach the SIEM and SOAR through the same channels as other security signals. A separate dashboard would hide them. Pairing classification with identity gives analysts one record of what each identity, human or agent, can reach. Traces use OpenTelemetry. Detections run on the live tool and API traces; warehouse access history arrives later and serves for reconciliation.

Give every control one owner and one enforcement point.

No component should silently recreate another component's decision. Each control has one system of record, an explicit enforcement point, a way to propagate policy and a tested failure behavior.

CONTROL	SYSTEM OF RECORD	ENFORCED AT	IF IT FAILS
Human identity	Enterprise identity provider	Platform, API gateway, source systems	Deny access
Agent delegation	Workload identity and agent runtime	Agent runtime, API boundary, each tool	Deny the tool call
Classification	DSPM tool and tagging workflow	Platform policies and retrieval filters	Redact, quarantine or deny
Entitlement and purpose	Enterprise authorization policy	Storage, source service, API and tool	Deny by default
Meaning	Governed ontology and semantic model	Generated and semantic views	Use the last approved version
Approval state	Workflow and change system	Provisioning and command workflow	Do not provision or execute
Audit evidence	Central observability and audit store	Every request and action	Alert, retain, investigate

Propagate policy like a deployment

Every policy integration declares its authoritative source, its target enforcement points, the propagation mechanism and expected latency, the reconciliation cadence and drift check, the rollback procedure and the owner of failed propagation. It also declares the behavior while policy state is uncertain. For sensitive data, that should fail closed.

The process and technology that hold it together.

Orchestration is the part most programs leave out. It turns three sets of components into one service you can change, monitor and recover.

ELEMENT	EXAMPLE	WHAT IT DOES
Control plane	One record of approvals, versions and policies	States what was approved, for which scope, under which conditions
Release pipeline	CI/CD with evaluation and policy gates	Every change to data, model, prompt, tool or policy passes the same gates
Pipelines	Apache Airflow coordinating dbt	Builds and tests the data and meaning layers on a schedule
Change and incident	ServiceNow	Change approvals, incidents and their link to the release record
Service levels	Objectives and error budgets	Freshness, latency, availability and recovery targets per capability
Runbooks	Tested by the receiving team	Restriction, rollback, revocation and recovery, run without the builders
Decision records	One per material component	Purpose, scope, owner, security boundary, exit strategy and replacement criteria

Technology choices implement bounded responsibilities. They aren’t the architecture. The decision record for each component is what lets a team replace it later without breaking the contracts around it.

TEST Ask the team to restrict one capability, roll back one release and revoke one user, using only the runbooks. Anything that needs a builder to step in isn’t transferable yet.

Name who owns each part of the system.

The architecture needs joint accountability. Three questions now share the same platform: what data means, how a person or agent reaches it safely and whether it becomes a live business capability.

GROUP	OWNS
Data group	The brain: data products, the ontology and semantic layer, and the contract every tool and skill must honor
Cyber group	The classification and tagging rules, identity and enforcement standards, and the SOC detections for agents
AI and digital product	The agents and the capability: scope, evaluation, release and user experience
Platform engineering	The shared substrate: environments, pipelines, observability and the control plane
Business owner	The workflow, the outcome measure and acceptance; supervises the agent like a manager

Self-service, in three tiers

Teams build in a sandbox on sample or masked data. They move to a managed sandbox on governed data with a named steward. They reach product grade only after certification and the readiness checklist in section 19. The gate between tiers is a governance decision. It isn’t a technical one.

CONVERGENCE The data group uses the classification the cyber group sets to declare a dataset ready for AI. Security stops being a late-stage blocker. It becomes what makes safe self-service possible.

An anti-money-laundering investigation assistant.

A bank's transaction monitoring produces alerts. Investigators assemble the customer profile, transactions, prior alerts, related parties and news coverage, then write a case narrative and decide whether to close or escalate. The assistant drafts the case summary. The investigator decides. Any suspicious activity report decision stays with qualified people.

SCOPE: READ ONLY · NO FILING · NO CUSTOMER CONTACT

DATA	LABEL	WHO MAY SEE IT
Customer and KYC profile; accounts and transactions	Customer financial data	Investigators assigned to the alert
Prior suspicious activity report records	SAR-restricted	The financial intelligence unit only
Related parties, from the ontology	Customer financial data	Investigators assigned to the alert
News and adverse media	External, unverified	Investigators; never treated as instructions

01 SECURITY The investigator signs in through single sign-on with MFA. The assistant runs under its own agent identity, registered to the head of the financial intelligence unit, with a read-only investigation role.

02 SECURITY Token exchange issues a short-lived token bound to this investigator and this agent, valid only for the case API, and limited to alerts in the investigator's queue.

03 AI The router recognizes an alert investigation and sends it to the AML agent, which plans the work: profile, twelve months of transactions, related parties, prior alerts and news.

04 DATA Profile and transactions come from certified data products through narrow read tools. Tag-based masking shows only the last four digits of tax identifiers.

05 DATA Related parties come from ontology-driven views, using the unit's approved definition of a related party. One possible match is flagged as unresolved.

06 SECURITY Prior SAR-related records carry the SAR-restricted label, so only financial intelligence unit roles see them, and the label stops that content from reaching any tool or channel outside the unit.

07 AI The draft summary cites every source record, separates facts from interpretation, and lists open questions for the investigator.

Where security earns its place.

The same request, and the events that test the controls around it.

01 SECURITY A news article retrieved for the case contains hidden text telling the assistant to close the alert as a false positive. The agent has no tool that can close an alert. The gateway flags instruction-like content, the attempt is logged, and the document is quarantined for review.

02 SECURITY The investigator asks for an email to the customer's relationship manager about the case. The draft carries the highest label among its sources, SAR-restricted, so data loss prevention rules block it from leaving the unit, and the assistant explains why. The event is logged.

03 SECURITY At 2 a.m. the delegated session starts requesting profiles for customers outside the investigator's queue. Each call is denied, the repeated denials fire a SIEM detection, a SOAR playbook suspends the agent identity and its sessions, and an incident opens.

04 SECURITY When the investigator moves to another team, the identity provider removes the unit role. The revocation test confirms the assistant stops returning restricted content within the tested window, including from the search index.

05 ORCHESTRATION The trace, the refused instruction, the blocked disclosure, the incident and the revocation test become release evidence. The owner decides whether to restore the agent, and on what conditions.

What reviewers will ask, and where the answers are

QUESTION	EVIDENCE
Which policy governs this assistant?	SR 26-2 places generative and agentic AI outside its scope, so the release record names the governing internal policy and its second-line reviewer
How is SAR confidentiality protected?	Banks may not disclose a SAR or information that would reveal its existence. The SAR-restricted label, role checks and loss-prevention rules are the controls; the blocked email is the test
Can you reconstruct this case summary?	The trace links every statement to its source record, version and policy decision

Measure the time to assemble case context, narrative quality under the unit's own quality review, rework and cost per completed investigation, against a baseline taken before launch. Illustrative; not a client deployment.

A claims assistant for life and disability claims.

A claims examiner needs to know who is connected to a claim, its current status and which procedure applies. The assistant gathers that context. The examiner makes every claim decision.

SCOPE: READ ONLY · NO ADJUDICATION · NO PAYMENTS · NO EXTERNAL CONTACT

DATA	LABEL	WHO MAY SEE IT
Claim status and history	Customer data	Examiners assigned to the claim
Medical and disability evidence	Medical information	Examiners with a medical review role
Group plan and employer data	Tenant: the employer	Examiners for that employer's plan
Procedures and guidance	Internal, by product and state	All examiners, filtered by product and jurisdiction

01 SECURITY The examiner signs in through single sign-on. The assistant runs under its own agent identity, registered to the claims owner, with a defined read-only role.

02 SECURITY The request carries a short-lived token bound to this examiner and this agent: only what both are allowed.

03 AI The router sends the request to the claims agent, which plans the task: status, related parties, and the applicable procedure.

04 DATA Claim status comes from a certified data product through a narrow API. Tag-based masking hides medical detail unless the examiner holds the medical review role.

05 DATA Related parties come from the ontology: insured, policyholder, beneficiary, employer or representative. One unresolved party match is flagged rather than guessed.

06 SECURITY Procedure documents were classified and labeled by the DSPM tool. Retrieval returns only the documents for this product and state that this examiner may read.

07 AI The answer separates facts from interpretation, cites its sources and names what the examiner still needs to confirm.

The controls under test.

The same request, and the events that test the controls around it.

01 SECURITY An attachment on the claim contains hidden instructions to email the file outside the company. The agent has no tool that can send externally; the attempt is refused, flagged and added to the attack tests.

02 SECURITY The claimant is also covered under a second employer's group plan. The tenant label keeps the second employer's records out of the answer, and the attempt is logged.

03 SECURITY Days later the agent starts requesting claims outside the examiner's assigned book. Each call is denied; the repeated denials fire a detection, and a playbook suspends the agent identity and its sessions and opens an incident.

04 SECURITY When an examiner leaves, the identity provider removes access, and the revocation test shows the assistant stops using that access within the tested window.

05 ORCHESTRATION The trace, the refused instruction, the tenant block, the incident and the revocation test become release evidence for the owner's decision.

What reviewers will ask, and where the answers are

QUESTION	EVIDENCE
How does this fit our AI program?	The NAIC Model Bulletin expects an insurer-wide AI program across the insurance life cycle, including claims; the system record shows how this capability fits it
Could it affect a claim outcome unfairly?	The examiner decides; evaluation includes checks for unfair outcomes wherever the assistant's output could influence one
Who could see medical information?	Role-based masking, its tests, and samples of allowed and denied requests

Measure the effort to assemble usable case context, including the examiner's verification time, repeat use and cost per completed task, against a baseline. Illustrative; not a client deployment.

Build it in phases, one capability at a time.

The full architecture is the destination. It isn't the first release. Each phase builds only the parts the chosen use case needs, and each ends in a decision.

PHASE	WHAT GETS BUILT	EVIDENCE FOR THE DECISION
Frame, 2–3 weeks	Use case, owners, target design, labels and data products in scope, review routes opened	A one-page alignment, a named owner, and the governing policy route
Prove, 4–6 weeks	A read-only slice: certified products, labels on those sources, one agent with two or three tools, its identity and SOC detections	Evaluation results, a request trace, a revocation test, and an injection test
Build, 8–16 weeks per release	Release pipeline and gates, policy propagation, runbooks, training and revised roles	Release record and tested runbooks
Run & Transfer	Your team operates it; adoption and cost are measured	Ownership demonstrated without the builders

The minimum first scope

One use case. The minimum certified data products. Labels on only those sources and documents. One read-only agent with its own identity and two or three narrow tools. Detections in the SOC for that agent. An evaluation set the business wrote, and a tested revocation. Everything else waits until the evidence says the next investment is worth making.

What you usually have already

Most enterprises already license a data platform, an identity provider, a SIEM and an IT service management tool. The work is usually connecting and governing them. Buying a new stack rarely helps. Each phase is separately scoped and priced in a written proposal before you commit.

A capability is production-ready only when every row is true.

Use this checklist at the release gate. A domain that can't meet every row isn't barred from building. It builds in a sandbox until it can.

REQUIREMENT	EVIDENCE OF COMPLETION
Named owner and support path	Owner, steward, security contact and escalation path in the manifest
Versioned contracts	Data product, API, tool and agent contracts reviewed and published
Identity and policy	Identity propagation and least privilege tested end to end, at the data
Labels	Every source classified; policy binding and cross-tenant denial tested at each approved serving path
Agent identity	Registered identity, owner, role, token policy and a tested kill switch
Monitoring	Traces reach the SIEM; agent detections and a SOAR playbook tested
Quality and evaluation	The evaluation suite passes in the release pipeline
Tool safety	Narrow scope, side effects declared, typed schemas, limits, and a security test pack covering prompt injection, tool confusion, privilege escalation and data exfiltration
Write safety, where needed	Command path with bound approval, operation IDs, reconciliation and outbox; the lost-response test passes
Operations	Service levels, alerts, runbooks, recovery and cost limits accepted
Reconstruction	One success and one failure reconstructed end to end from the trace

RULE No agent tool is production-ready just because it can be called successfully from a chat window or a development tool.

Sources, checks and disclosure.

Prepared by Richard White, October 2026, from his reference architecture. Product capabilities change quickly: each product fact below was checked against vendor documentation or announcements in October 2026, and should be re-checked before you rely on it. Each source is linked; links updated 15 October 2026.

Products Snowflake documentation on tag-based policies: masking generally available, tag-based row access in public preview, tag propagation to downstream objects. Snowflake release note, 28 August 2026, recommending Cortex Agents over Cortex Analyst. Databricks, 13 May 2026: attribute-based row filters and column masks, governed tags and data classification generally available in Unity Catalog. Microsoft Learn: Microsoft Entra Agent ID, generally available, and its on-behalf-of flow, which uses the JWT bearer grant. Snowflake: Cortex Search services run with owner's rights. Cyera announcements: Omni DLP (April 2025), AI Guardian (August 2025), Agent Guardian (August 2026). ServiceNow's acquisition of data.world, announced May 2025.

Standards Model Context Protocol, Security Best Practices, 25 November 2025. IETF RFC 8693, OAuth 2.0 Token Exchange; RFC 8707, Resource Indicators for OAuth 2.0. OpenTelemetry semantic conventions for generative AI, still in development. OWASP Cheat Sheet Series on prompt injection and AI agent security; OWASP Top 10 for Agentic Applications (2026). Amazon Builders' Library, "Making retries safe with idempotent APIs."

Regulation Federal Reserve, OCC and FDIC, SR 26-2, Revised Guidance on Model Risk Management, 17 April 2026. 31 CFR 1020.320(e), confidentiality of suspicious activity reports. Gramm-Leach-Bliley Act. NAIC Model Bulletin on the Use of Artificial Intelligence Systems by Insurers, December 2023. NYDFS 23 NYCRR Part 500. See *AI Governance Evidence* for the full regulatory treatment.

Disclosure The author has worked closely with several vendors named here, including Cyera, ServiceNow, Snowflake, Databricks and Microsoft, as a customer and practitioner. No vendor reviewed or paid for this paper. Products are named as examples of the patterns, not endorsements; substitutes must keep the same contracts and controls.

The two use cases are illustrative. They aren't client deployments or achieved results. Not legal advice.

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.

ARCHITECTURE **The Four-Part System**
The same architecture, vendor-neutral, in 16 pages.

GOVERNANCE **AI Governance Evidence**
What US bank and insurance reviewers inspect, in 18 pages.

TECHNICAL **Agent Onboarding**
The agent and tool contracts behind section 09, in 13 pages.

RJL SOLUTIONS

WEB rjlsol.com

EMAIL rwhite@rjlsol.com

This paper describes no client engagement and no client result. The use cases are illustrative, and the products are examples, not endorsements. Any implementation has to validate platform capabilities, licensing, regulatory obligations and operational ownership for the actual environment.