RJL SOLUTIONS

OWNERSHIP AND DELIVERY

The Data and AI Operating Model

A practical model for business ownership, stable delivery teams and shared enterprise controls.

CLEAR OWNERSHIP · REPEATABLE CONTROLS · OPERATING EVIDENCE

Richard White
RJL Solutions · rjlsol.com

ARCHITECTURE UPDATE
29 OCTOBER 2026

01 / THE EXECUTIVE CASE

Organize around outcomes.

No platform investment can fix unclear business ownership, or a delivery queue that grows faster than capacity.

An insurer brings four policy-in-force counts to a leadership meeting. Some of the differences are legitimate, because servicing, finance and actuarial analysis use different dates or inclusion rules. Others come from undocumented transformations. Four numbers aren't the real problem. Nobody can explain which definition serves which decision, reconcile the differences or authorize a change. That's the problem.

An effective operating model splits the work three ways. Business domains take responsibility for meaning and outcomes. Delivery teams get persistent capacity. A shared platform takes responsibility for repeatable services and controls. A central data office sets common expectations and owns selected enterprise products. It doesn't become the approval queue for every change.

LEADERSHIP DECISION	EVIDENCE TO ASK FOR
Who owns the outcome?	One accountable business owner; defined purpose and measures; a route for resolving disputes.
Where does delivery capacity sit?	A named team that builds, operates and improves the product, with a funded backlog.
What should be shared?	Reusable services with owners, contracts and service commitments; adoption by domain teams.

START WITH A BOUNDED PROBLEM	Pick a material decision or workflow. Baseline its delay, failure and operating cost. Change ownership and delivery around that outcome, then scale the parts that prove useful.

This paper connects organizational choices to implementation. It covers teams, decision rights, funding, a staged transition and the measures that show whether the model works.

02 / ORGANIZATIONAL CHOICE

Choose the shape deliberately.

Federation helps when the business can own meaning and sustain delivery. It's no reason to duplicate every central capability.

SHAPE	USEFUL WHEN	FAILURE TO WATCH
Centralized	Skills are scarce; domains need substantial support; common data needs dominate.	The central team becomes a queue and definitions become remote from the business.
Decentralized	Domains have strong delivery capacity and substantially independent needs.	Tools, controls and definitions diverge; consumers integrate the fragments.
Hub-and-spoke	A transition needs shared specialists and platform services with domain-facing leads.	Spokes depend on hub approval for ordinary work; accountability remains ambiguous.
Domain product teams + platform	Domains can own products and outcomes; reuse and common controls matter.	Domains are nominally accountable but lack people, funds or authority.

These shapes can coexist. A small enterprise may start centrally, put delivery leads in two domains and keep specialist engineering central. A larger one may run standing domain teams for claims or lending while keeping enterprise identifiers and shared infrastructure under common ownership.

Test the prerequisites before choosing federation: available domain leadership, sustainable engineering capacity, product boundaries, common identity and policy services, and support for operating products after release. Where any are missing, build them through a supported transition.

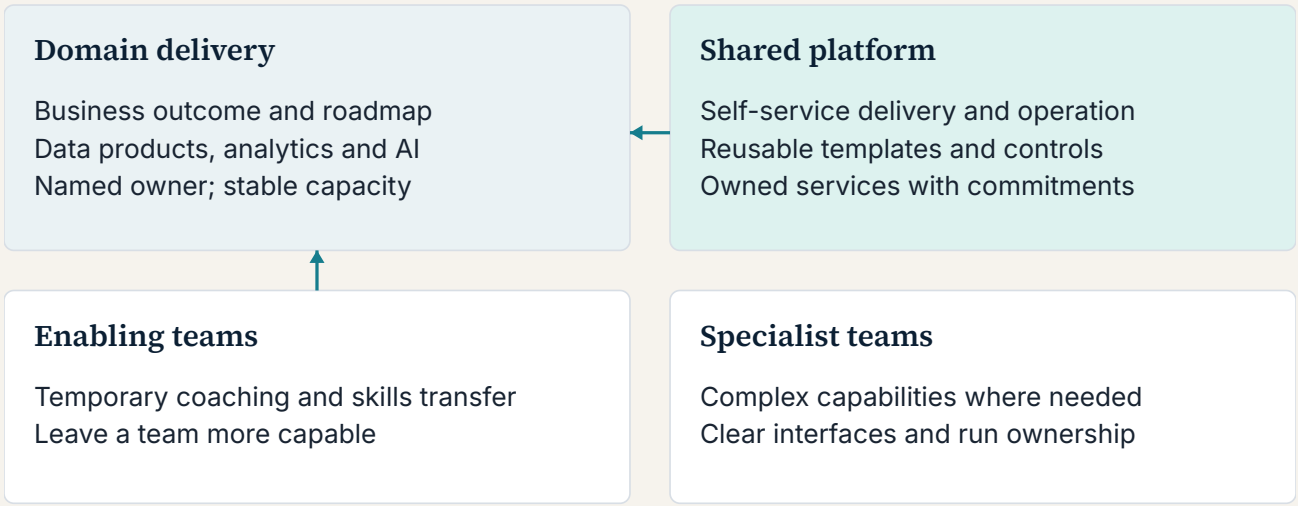
AN EXPLICIT TRADEOFF

Domain capacity costs more than a paper reorganization. Fund the smallest viable delivery model and shared platform, and remove duplicated work as adoption grows.

03 / TEAM DESIGN

Separate delivery from capability building.

Stable teams own products. Shared services reduce repeated work. Professional disciplines develop the people doing it.



Professional disciplines support standards, hiring and careers.

RJL team design, informed by the team types described in Team Topologies [1].

A domain team owns a business-aligned backlog and carries its products through operation. The platform team owns services that domains consume. Enabling teams transfer skills through time-bounded coaching. And a specialist team may own a technically complex capability, where concentrating expertise is justified.

Professional disciplines, or chapters, provide hiring standards, development, engineering practices and career support. They're a permanent professional home. That's different from an enabling team. They shouldn't create a second delivery backlog, or take work prioritization away from the domain team.

AVOID THE PROJECT STAFFING LOOP	Keep a core team together long enough to learn the domain and operate what it builds. Borrow specialist capacity for defined needs. Don't rebuild the whole team for each request.

04 / PLATFORM AS A SERVICE

Make the shared path useful.

A platform earns adoption by making delivery easier while carrying common controls.

TEAM NEED	PLATFORM SERVICE	EVIDENCE OF USEFULNESS
Find	Catalog, definitions, owners and current product status.	Time to locate an appropriate product; failed searches.
Get access	Identity, policy decisions, request routes and scoped credentials.	Time to usable access; unjustified manual exceptions.
Build	Templates, integration patterns, approved runtime and test tools.	Time to a first working increment; template adoption.
Publish	Contracts, quality checks, lineage and release automation.	First-pass release success; defects after publication.
Operate	Monitoring, incident response, recovery and cost visibility.	Service commitment breaches; recovery time; cost per use.

Give each service an owner, a roadmap, a supported interface and a service commitment. A catalog with poor metadata, or a template that can’t deploy into the existing environment, isn’t a useful platform service just because it exists.

Hold regular feedback sessions with domain teams. Tell an unpopular service apart from a required control: improve the delivery experience while keeping the policy obligation. Track approved exceptions, their compensating controls and their expiry.

WHAT CENTRAL LEADERSHIP STILL OWNS	Enterprise standards, common identifiers and reference products, shared policy authority, major platform investment and cross-domain dispute resolution remain explicit central responsibilities.

05 / DOMAIN DELIVERY

Give meaning a named lead.

Product priority, technical delivery and business meaning need clear responsibility, whether or not each requires a separate job title.

RESPONSIBILITY	WHAT THE ROLE DOES
Business data owner	Accountable for meaning, permitted business use, material quality risk and the product outcome.
Product owner	Prioritizes consumer needs, backlog, adoption and product lifecycle within the agreed outcome.
Technical lead	Owns engineering choices, delivery quality and reliable operation within enterprise constraints.
Data lead / domain architect	Connects domain strategy to definitions, product boundaries, reuse and enterprise coherence.
Steward	Maintains definitions, metadata and quality rules; coordinates issue resolution and evidence.

The data-lead responsibility earns its place when a domain has several products or competing definitions. It shouldn't duplicate the product owner or become another approval layer. In a smaller domain, an existing architect or senior data practitioner can carry it, with protected capacity and a clear mandate.

In the first three months, expect an agreed domain boundary, critical definitions and a credible first product contract. By six to twelve months, look for reuse, resolved definition disputes and a consistent portfolio. Over the longer term, the evidence is fewer duplicate products and less coordination for routine changes.

Team size follows the workload and the support model. A core group may combine product, engineering, stewardship and analytical skills, backed by platform and specialist teams. Reserve capacity for incidents, maintenance, documentation and consumer onboarding.

ASSIGN TIME AS WELL AS NAMES	A named owner without authority or time is just a routing label. Agree decision deadlines, escalation and operational cover.

06 / DECISION RIGHTS

Make the boundary of authority visible.

Use one accountable owner for each decision. Contributions and independent challenge remain explicit.

DECISION	ACCOUNTABLE ROLE	CONTRIBUTORS / CHALLENGE
Business definition and intended use	Named business data owner	Stewards, consuming functions; central data office resolves enterprise conflicts.
Product priorities and value	Named business sponsor	Product owner and consumers; finance tests the business case.
Engineering and runtime choices	Named technology executive or delegate	CIO/CTO teams, domain technical leads and platform owner.
Data standards and cross-domain coherence	CDO/CDAO or delegated data authority	Domain data leads; architecture and business owners.
Security policy and access controls	Security policy authority	CISO team, platform engineering, privacy and business owners.
Risk policy and independent challenge	Enterprise risk authority	CRO/compliance/model risk, with appropriate independence from delivery.

Organizations divide CIO, CTO, CDO and CDAO responsibilities differently. Settle the allocation explicitly; the titles don’t settle it for you. For shared engineering capabilities, name one accountable technology owner and document the services delivered to other executives.

For a cross-domain product such as claim cost to close, name one accountable business owner. Finance may approve reconciliation rules, and claims may supply source semantics. Joint participation doesn’t require anonymous joint accountability.

DEFINE THE ESCALATION Record which decisions the team can make, which need approval, who resolves a dispute and how fast. Escalate in proportion to materiality.

07 / GOVERNANCE AND RISK

Carry evidence through delivery.

Enterprise oversight should be able to see the outcome, exposure and control evidence without reconstructing the product after release.

Boards and executive committees set direction and risk appetite. Business management owns outcomes and first-line controls. Risk and compliance functions provide policy and independent challenge as appropriate, and internal audit provides independent assurance. The product team and platform produce evidence these functions can inspect.

RISK QUESTION	EVIDENCE IN OPERATION
Can we explain the result?	Versioned definitions, source-to-output lineage and reconciliation of critical measures.
Can we control access and use?	Approved purposes, scoped identity, tested enforcement and exception records.
Can we recover?	Monitored commitments, incident ownership, tested fallback and recovery evidence.
Can we govern an agent?	Tool contracts, evaluations, approval gates, traces and a means to stop or restrict it.

In April 2026, the Federal Reserve, OCC and FDIC issued revised model-risk guidance, referenced by the Federal Reserve as SR 26-2, replacing SR 11-7 and SR 21-8. It's nonbinding guidance, most relevant above $30 billion in assets, though smaller banks with significant model risk may also find it relevant. Generative and agentic AI fall outside its stated scope. That doesn't remove the need to govern their risks. The guidance directs banks to determine appropriate governance and controls for tools outside scope [4].

Treat model risk, data risk, information security, privacy, third-party risk and operational resilience as connected responsibilities. Insurers should confirm applicable state and other supervisory expectations with the responsible risk and legal functions. A generic control pattern doesn't establish compliance.

A growth company or private-equity portfolio company without a second line or model risk function still needs the same answers. Name an executive who can challenge the release, such as the CFO or an outside adviser. Scale the evidence to what the tool can change. Diligence, financing and an eventual sale will all ask the same questions.

PROPORTIONATE, INDEPENDENT CHALLENGE	Define the evidence and approval each use case needs from its risk. A low-impact assistant and a tool that changes financial or customer outcomes shouldn't pass through the same undifferentiated release gate.

Build governance from the center out.

Governance works as four rings, and each ring depends on the one inside it.

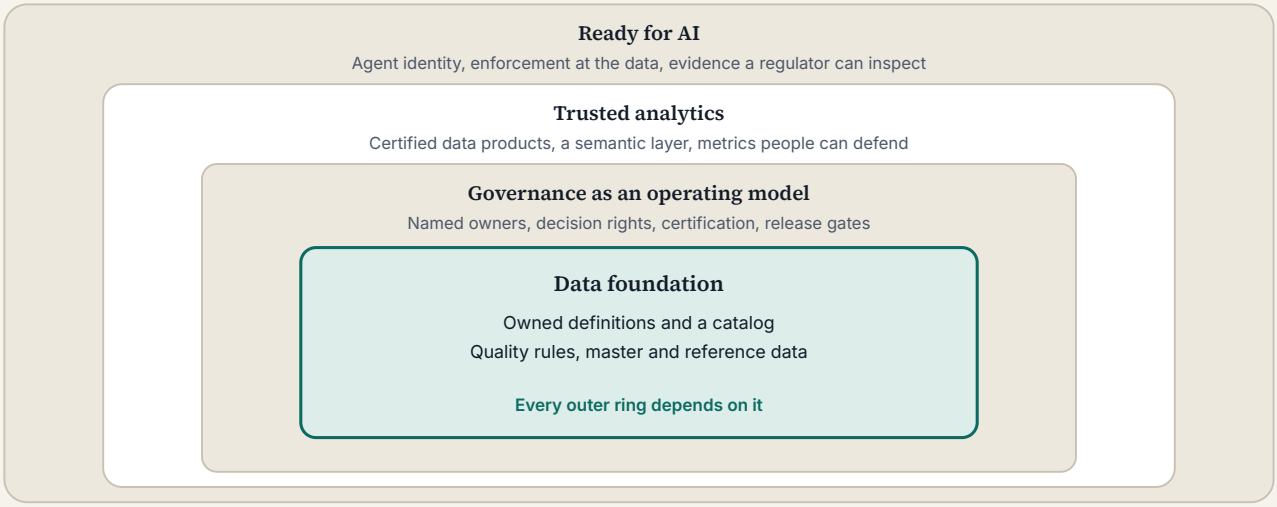


Exhibit: Governance is built from the center out. The outer ring cannot be bought on its own.

Start with the data foundation: owned definitions, a catalog, quality rules, and master and reference data. Governance as an operating model adds named owners, decision rights, certification and release gates. Trusted analytics and readiness for AI build on both.

A tool bought for the outer ring can’t supply what the inner rings haven’t established. Sequence investment from the center. Judge each ring by the evidence it produces in operation, not by the policies written for it.

08 / INTERACTION AND FUNDING

Fund enduring capability.

Standing capacity and clear team interfaces cut the handoffs that project-by-project staffing keeps recreating.

INTERACTION	USE IT FOR	EXIT CONDITION
Collaboration	A domain and platform team jointly establish a new product or delivery pattern.	A tested interface and agreed ownership allow independent work.
Service consumption	A domain uses stable catalog, access, runtime or publication services.	Service levels and support paths remain useful and visible.
Facilitation	A coach helps a team learn stewardship, quality testing or another capability.	The team can perform the activity without permanent coaching.

The interaction modes build on Team Topologies [1]. Choose the mode on purpose. Permanent collaboration for routine requests usually signals a weak interface or a missing self-service capability.

Fund core teams as capacity, with a business-owned backlog. Set aside explicit capacity for running products and reducing technical debt. Use separately approved investment for major expansion, new risk exposure or a costly platform capability.

Make platform costs visible through showback before introducing complex internal charging. Distinguish shared foundations from consumption-driven costs. Track whether reuse actually avoids duplicate build and run expense. Don’t assume every shared service lowers total cost.

A USEFUL QUARTERLY REVIEW Review outcomes, adoption, service failures and cost together. Increase, redirect or reduce capacity on the evidence. Never protect an unused product just because its original project closed successfully.

09 / TRANSITION

Prove, extend, make it default.

A first governed capability and an enterprise operating-model transition are different scopes.

PROVE

0-3 months
One bounded domain
First governed product
Baseline and adoption

→

EXTEND

3-9 months
Reuse across domains
Expand shared services
Retire duplicate paths

→

MAKE IT DEFAULT

9-18 months
Standing team capacity
Broad control coverage
Evidence-led funding

Indicative enterprise roadmap; progression depends on exit evidence.

Indicative planning horizons. These are not delivery guarantees or a substitute for a scoped engagement.

PHASE	EXIT EVIDENCE
Prove	An accountable domain, a functioning delivery team, a used product and tested minimum controls.
Extend	Another team reuses the product or delivery pattern; shared services meet commitments; redundant work is identified.
Make it default	Budget, career structures, routine request routing and risk oversight support the operating model.

The work overlaps. Inventory products and decisions while building the first capability. Improve access as consumers arrive, and consolidate only once you understand the dependencies. Progress depends on leadership, existing controls, data, procurement and consumer readiness. Keep named decisions in the transition backlog; elapsed time isn't evidence of readiness.

PROTECT THE FIRST RELEASE	Classification, authorized access, critical quality tests, essential lineage, monitoring and incident ownership are prerequisites. Broader automation can come later. Basic control evidence can't.

Measure value, reliability and delivery.

A useful scorecard exposes the tension between faster delivery, controlled operation and business benefit.

MEASURE	PRACTICAL DEFINITION
Business outcome	Change in decision time, manual work or error for the named workflow; use an agreed baseline.
Time to first usable product	Elapsed time from an accepted need to authorized, working consumption.
Reuse and duplication	Supported use beyond the first; duplicate build and run paths retired without breaking consumers.
Reliability and trust	Missed freshness/quality/availability commitments; downstream impact and time to restoration.
Adoption	Eligible consumers actively using the product, with denominator and observation period stated.
Economics	Build and run cost, cost per relevant use and measured avoided expense.
Consumer experience	Access delay, support burden and satisfaction by consumer type.

Where software delivery metrics help, adapt DORA with clear definitions: change lead time, deployment frequency, failed deployment recovery time, change fail rate and deployment rework rate [3]. Apply them to a product or service. A rise in deployment frequency alone says nothing about business value.

Compare similar workloads, and report distributions where averages hide the extremes. Separate experimentation from material production failures. Link service incidents to the decisions or consumers they affected, and use the evidence to prioritize improvement.

NO INVENTED BUSINESS CASE Set targets after you've measured the baseline. Reuse counts and conceptual cost curves are design arguments. They don't prove savings.

11 / ILLUSTRATIVE APPLICATION

An insurer changes the delivery loop.

Claims leadership wants examiners to spend less time assembling case information while preserving decision accountability.

The sponsor defines a bounded examiner-assistant use case. It retrieves permitted case information and current procedure guidance, cites the sources and prepares a summary. It doesn't approve payments or make coverage decisions. Claims names a data owner and a product owner; technology supplies a stable engineering team and a supported runtime.

The team publishes claim status and payment history as separate products, each with the right interface and freshness commitment. Finance approves payment reconciliation rules. Claims stays the accountable owner of claim products. A separate claim-cost aggregate has one named business owner and a versioned definition.

DELIVERY CHOICE	EVIDENCE TO RETAIN
Start with one examiner workflow	Baseline preparation time, eligible users and manual verification steps.
Use existing platform services	Authorized identity, product contracts, policy tests, monitoring and support.
Put context before retrieval	Current document rights, source versions, case scope and denial tests.
Release through an agreed gate	Representative evaluations, negative access tests, fallback and accountable approval.
Review after adoption	Use, overrides, incorrect summaries, incidents, time saved and full operating cost.

The first release establishes a pattern. It isn't an enterprise transformation. Expansion into another product line follows only when that line can support ownership, semantics and control evidence. Policy definitions that differ for different purposes stay different, and get reconciled where comparison requires it.

ILLUSTRATIVE, NOT A CLIENT RESULT	This example describes design choices. It claims no realized savings, implementation duration or client deployment.

Turn responsibilities into services.

Use this inventory to assign the work that often disappears between organization charts.

CAPABILITY	NAMED RESPONSIBILITY	FIRST EVIDENCE
Catalog and discovery	Platform service owner; domain metadata owners.	A consumer finds a current product and its supported contact.
Reference products	Accountable owner for identifiers, codes and hierarchies.	Change process, stewardship and downstream compatibility tests.
Policy and identity	Policy authority; platform enforcement owner.	Permitted and prohibited paths tested with scoped identities.
Integration and templates	Platform owner and domain engineering leads.	A reproducible pipeline or API path with release checks.
Enablement and disciplines	Coaching owner; professional discipline leads.	Capability transfer plus hiring and development standards.
Run and recovery	Product run owner; shared operations service.	Alerts, escalation, fallback and recovery tested.

First-quarter decisions

- Name the business sponsor, accountable data owner and delivery lead for one outcome.
- Agree a product boundary, supported interface and minimum release evidence.
- Confirm capacity for build, operation, consumer onboarding and risk review.
- Set the dispute, exception and incident routes before the first release.
- Baseline value, reliability and cost; decide what evidence justifies expansion.

Use the checklist to expose gaps. Don’t use it to create more committees. Keep the responsibility allocation small enough for the team to explain, and specific enough for an executive to act on.

PRODUCT CONSTRUCTION / PRODUCT REVIEW

Fund the supported service, not just its launch.

Use a short, evidence-based review of a product and its consumers. This is an operating decision, not a proposed organization chart.

DECISION	EVIDENCE AND ACCOUNTABLE RESPONSIBILITY
Value and demand	Business owner reviews the supported decision, eligible consumers, actual adoption and measured outcome. Stop treating catalog entries as delivered value.
Meaning and composition	Domain owners approve definitions; one owner answers for the derived receivables definition and source agreements. Preserve legitimate purpose-specific meanings.
Reliability and control	Engineering/run owners review source completeness, reconciliation, access denials, service breaches, incidents and recovery evidence.
Capacity and funding	Product owner accounts for support, consumer onboarding, compatibility, platform dependencies and change backlog, not only initial build cost.
Next action	Agree expand, repair, consolidate, hold or retire. Record authority, cost, consumer impact and the evidence required at the next review.

For a second receivables consumer, ask whether the existing definition, rights and cutoff fit. If they do, fund supported onboarding. If not, agree a versioned derived interface or a separate product with a clear owner. Do not force incompatible uses into one universal definition to claim reuse.

Individuals may hold several responsibilities in a small firm; regulated uses may need independent challenge. Allocate decision authority explicitly without assuming a particular reporting structure. See Data Products in Practice and AI Token Economics for the shared worked example and cost categories.

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.

Sources and scope.

Public frameworks inform the design. The implementation choices, checklists and examples are RJL recommendations, drawn from Richard White's practice.

[1] *Team Topologies*. Skelton and Pais; core team types and interaction modes. The organizational allocation in this paper is an RJL adaptation. teamtopologies.com / core team types

[2] *Data Mesh Principles and Logical Architecture*. Zhamak Dehghani, December 2020. Domain ownership, products, self-service platform and federated governance. martinfowler.com / data mesh principles

[3] *DORA delivery performance metrics*. Current five-metric model. Product value and data service measures in this paper are additional RJL recommendations. dora.dev / delivery performance metrics

[4] *Revised model-risk guidance*. Federal Reserve SR 26-2, 17 April 2026; interagency guidance also published by the OCC. Scope is not a general exemption from AI governance. federalreserve.gov / SR 26-2

[5] *Model Risk Management: Revised Guidance*. OCC Bulletin 2026-13 attachment; scope and governance discussion, including generative and agentic AI. occ.gov / revised guidance

Sources reviewed 10 October 2026; each is linked. Examples are illustrative; proposed measures and service commitments need an agreed baseline and risk assessment. This paper does not establish regulatory compliance or reproduce a licensed assessment standard. Companion papers at rjlsol.com/whitepapers.

Read the series

PAPER	WHAT IT COVERS
The Data and AI Operating Model (this paper)	Ownership, teams, decision rights and funding.
Data Products in Practice	Boundaries, contracts, quality and certification.
The Federated Data Platform	Shared services, access paths, resilience and agents.
Context Before Query	Context, authorization and enforceable policy for AI.
AI Governance Evidence	For regulated firms: what bank and insurance reviewers need to inspect.

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.

Bring one stalled use case, one disputed business measure and the current delivery structure. RJL Solutions helps define accountable ownership, a workable team model and the evidence needed to move into operation.

PRODUCTS **Data Products in Practice**
Boundaries, contracts, quality and certification.

PLATFORM **The Federated Data Platform**
Shared services, access paths, resilience and agents.

GOVERNANCE **Context Before Query**
Context, authorization and enforceable policy for AI.

RJL SOLUTIONS

WEB rjlsol.com

EMAIL rwhite@rjlsol.com

Examples are illustrative and describe no client engagement or client result. Measures, timings and service commitments are starting points that need an agreed baseline. Apply the design to your business purpose, existing controls and any obligations that apply to you.