RJL SOLUTIONS

PRODUCTS AND TRUST

Data Products in Practice

Define products people and systems can use, with clear meaning, ownership and operating commitments.

CLEAR OWNERSHIP · REPEATABLE CONTROLS · OPERATING EVIDENCE

Richard White
RJL Solutions · rjlsol.com

VERSION 3 / EXPANDED GUIDE
29 OCTOBER 2026

01 / THE EXECUTIVE CASE

A product is a promise to its consumers.

Publishing a table establishes nothing about its meaning, supported use or reliability.

A consumer should be able to find a product, understand what it represents, get authorized access and use it through a supported interface. Behind it, a team maintains the contract, monitors commitments and manages change. That continuing responsibility is what separates a data product from an unmanaged extract.

The product approach applies to people, applications, analytical models and agents alike. A payment-history product and a live claim-status API can both be products. They need different freshness commitments and operating designs, though. Reuse is valuable. Still, a critical product serving a single application can merit permanent product management.

CONSUMER QUESTION	EVIDENCE OF A USABLE PRODUCT
Can I find and identify it?	Catalog entry, stable identifier, supported contact and lifecycle status.
Can I understand it?	Purpose, grain, definitions, units, dates, valid joins and limitations.
Can I obtain and use it?	Authorized interface, access route, examples and known dependencies.
Can I rely on it?	Quality and freshness commitments, monitoring, provenance and support.
Can I absorb a change?	Version policy, compatibility expectations and retirement notice.

THE LEADERSHIP DECISION — Choose the business outcome and the accountable owner first. The product label comes later. Fund maintenance, consumer support and change management as part of the product.

Data mesh informs the usability principles [1]. The contracts, tests and implementation choices that follow are practical recommendations for delivery in regulated enterprises.

02 / PRODUCT, NOT PROJECT

A relabeled dataset is still a dataset.

A project optimizes for delivery. A product optimizes for adoption, reuse and value that lasts after the delivery team moves on.

Many programs rename existing tables as products and publish them to a catalog. Without a defined consumer, an accountable owner and a promise someone maintains, you get an artifact with a new label. Here's the test. Could a named consumer rely on it tomorrow without asking the team that built it?

Confirm demand before you build

Validate the need before investing in pipelines. A draft contract, a sample with real definitions, a mock interface or a catalog listing can show whether consumers will use the product, and what it must promise them. A short conversation with the people whose work would change costs far less than a well-governed product nobody adopts.

ILLUSTRATIVE EXAMPLE

A finance function publishes dozens of products to an internal marketplace. Each is cataloged, governed and tested. Adoption stays low and teams keep their spreadsheets, because no one confirmed which decisions the products were meant to serve. Engineering quality was never the problem; demand was.

Make reuse the easier path

Teams duplicate data when they're measured on local delivery and reuse costs them time. Make the shared product easy to find and consume. Support its first adopters. And recognize both sides: the producer who publishes for reuse, and the consumer who adopts instead of rebuilding.

Hold the owner to adoption and value

Ownership doesn't end at publication. The owner answers for adoption, consumer experience and value as well as quality. They use the product measures later in this paper to decide whether to enhance, consolidate or retire it. Governance should ask whether a product creates value, not only whether it complies.

Keep business meaning within a clear boundary.

A domain is a responsibility boundary. A product is a supported asset published within or across those boundaries.

The word account can mean a deposit account, a loan account or a general-ledger account. Treat those meanings as interchangeable and you get incorrect joins and disputed measures. A bounded context keeps the meaning consistent within a purpose, and defines how it maps to other contexts [2].

BOUNDARY TEST	WHAT TO DECIDE
Who can approve a definition change?	Name the business authority; split unrelated approval responsibilities.
Who owns a material failure?	Align incident and quality responsibility with a capable owner.
Do consumers need the same grain and history?	Separate incompatible latency, retention or interpretation requirements.
Can an interface express the relationship?	Use a versioned contract instead of merging unrelated responsibilities.

Start with a domain charter: scope, critical concepts, source responsibilities, published products and the route for disputes. A governance forum can maintain the charter. It can’t substitute for engineering and operational capacity, so identify the delivery team separately.

A customer-360 or policy-file view can be a useful consumer-facing composition over smaller products. It shouldn’t silently absorb ownership of consent, claims, billing and finance. Publish the component contracts, and record the aggregate definition.

SMALL ENOUGH TO CHANGE, SUBSTANTIAL ENOUGH TO RUN	Small products can reduce coordination. Merging them isn’t automatically cheap, though. Account for consumer migration, compatibility, authorization and ownership before changing a boundary.

Change the data, take on the contract.

The producer closest to the business owns the product. A consumer who changes it and shares the result becomes a producer.

ROLE	OBLIGATIONS
Producer: the domain closest to the business meaning	Owns the product and its contract: meaning, quality, freshness, access and support. Corrects errors at the source.
Consumer who reads	Uses the data within the contract terms: purpose, classification and retention. No other product obligations.
Consumer who changes and shares	Becomes the producer of a derived product, with a named owner, definitions, tests, service levels, access rules and support for what it introduces.

Rules that hold across the chain

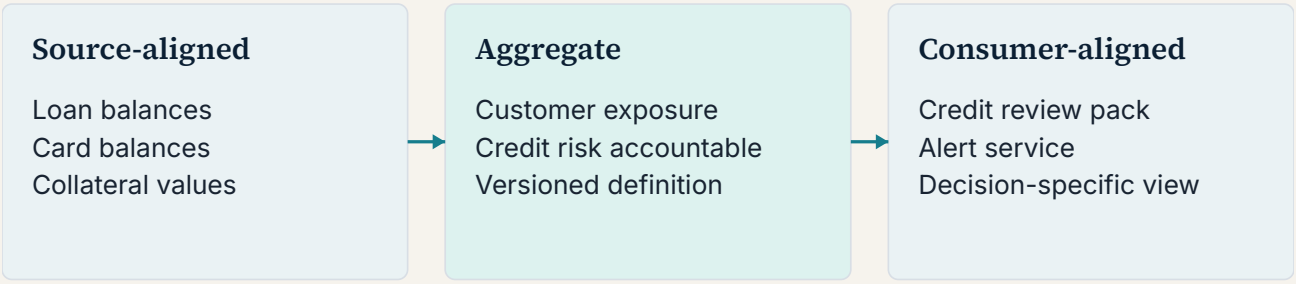
- Restrictions travel downstream. A derived product inherits the source classification, purpose limits and retention. It can’t loosen them except through an approved, recorded de-identification or aggregation decision.
- Accountability doesn’t flow upstream. The source producer isn’t accountable for a consumer’s logic, and certifying the source doesn’t certify the derived product.
- Correct reusable data at its authoritative source. A consumer reports an error to the producer. Any temporary local correction records its basis and expiry, and gets reconciled once the source is fixed.
- Obligations start when others rely on the result. A private analysis may not need a published product contract. Source permissions, terms and validation matched to its consequences still apply. Share a result that others rely on, and the producer obligations begin.

WHY OWNERSHIP BELONGS CLOSE TO THE BUSINESS	Only the people who can change the process, the policy or the customer promise can keep a contract honest. Own a product far from that business and it drifts toward whatever is easiest to build.

05 / PRODUCT TYPES

Describe the responsibility, then the delivery.

Use a primary classification for ownership and separate dimensions for architecture and service commitments.



Each published product has an owner, a contract and a supported interface.

Illustrative product composition. Terms are an implementation adaptation informed by data mesh [1].

PRIMARY TYPE	ACCOUNTABILITY
Source-aligned	A business domain owns a faithful, usable representation of source activity.
Aggregate	One named owner answers for a combined definition, dependencies and reconciliation.
Consumer-aligned	The consuming function owns a purpose-specific view or interface and its outcome.

These types describe responsibility and composition. They say nothing about whether the product is analytical or operational, current or period-end, structured or document-based. A consumer-aligned product may serve one critical application and still be far more than a temporary project.

ONE ACCOUNTABLE OWNER Claims and finance may jointly define claim cost to close. Name one accountable owner anyway, with finance responsible for specified reconciliation or approval. Never leave a product owned by a committee.

06 / DELIVERY TYPES

Operational, analytical or knowledge: design each differently.

Delivery type is independent of the responsibility type above. Foundational and derived levels apply within each.

DELIVERY TYPE	THE PRODUCT IS	SERVICE LEVELS TO STATE
Operational, including transactional	A contracted API, event stream, change feed or serving store, on the system of record or as a serving copy of gold. Writes use a separate, approved command path.	Latency, availability, ordering and behavior when the source fails
Analytical	The gold table or view: gold is the publishing convention for analytical products. In this reference design, bronze and silver stay inside the pipeline; other publication conventions can be valid under a supported contract.	Refresh cadence, completeness, period represented and finality
Knowledge	A governed document collection or index that carries source, version, sensitivity label and access list.	How fast deletions, revisions and permission changes reach answers

Two levels within each type

LEVEL	OPERATIONAL EXAMPLE	ANALYTICAL EXAMPLE	KNOWLEDGE EXAMPLE
Foundational: close to the source	Claim status events from the claims system	Order lines in gold, one row per line	Approved policy documents with their access lists
Derived: combined for a purpose	A next-best-action signal for service agents	Net sales by channel and claim cost to close	A curated corpus for one assistant, with retired versions removed

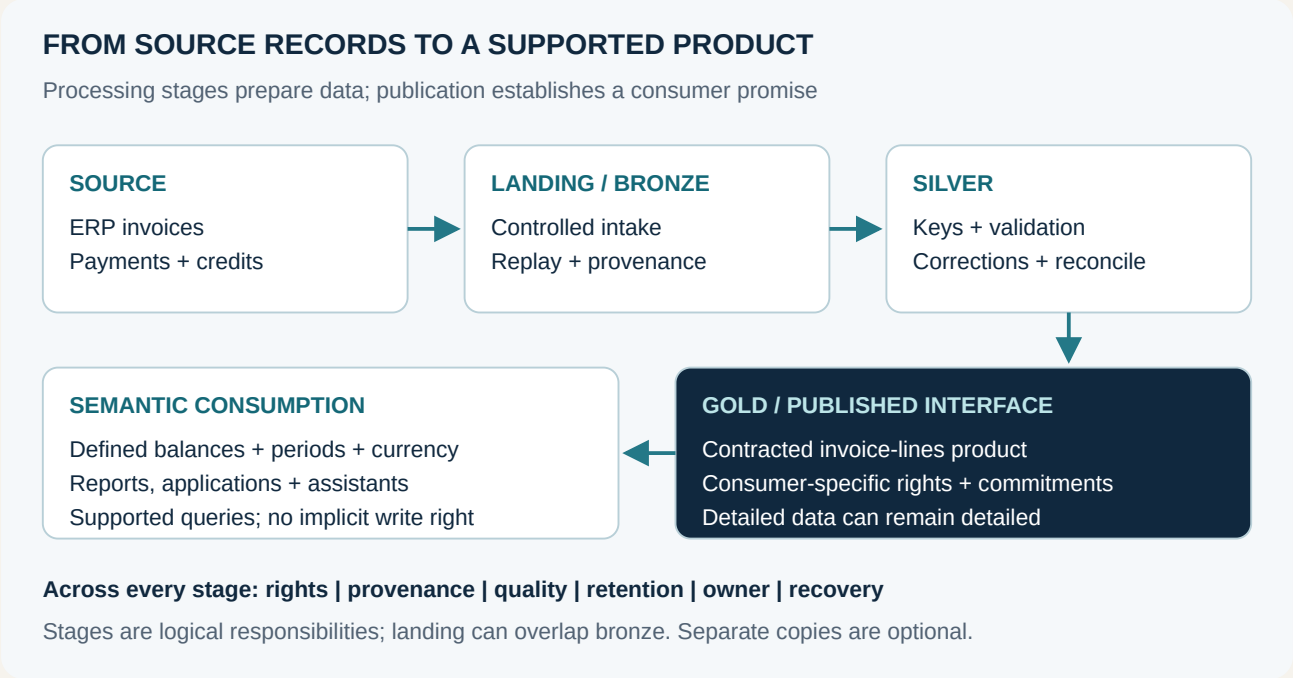
A product is defined by its contract, owner, service levels and access rules. Where it's stored doesn't matter. A transactional system becomes a data product when it publishes such an interface. Letting other teams query its internal tables doesn't make it one.

Classic data mesh treats data products as analytical and keeps operational APIs as a separate interface [1]. This paper treats a contracted operational interface as a product too, because consumers depend on it in the same way. Foundational doesn't mean untouched source data. Derived doesn't necessarily mean cross-domain. Record transformations, grain and provenance; don't rely on the label.

CHOOSE THE SERVING PATH Analytical needs use warehouse or lakehouse queries on gold. Live state uses APIs, event feeds or serving stores. Documents for AI use a permission-aware index. All three need ownership, tests, lineage and controlled access.

Follow the records, then define the promise.

Synthetic example: Finance needs a reliable receivables report, and an assistant needs to explain the same dated balance.



RJL analytical reference path. Gold is the publishing convention used here; it is not a universal certification rule.

Start with the consumer question, invoice grain, legal entity, currency and as-of time. Identify the authoritative invoice, credit and payment sources. The stages prepare those records for supported use; they do not automatically create demand, ownership or authorization.

This example keeps landing and bronze together when one controlled intake service can meet both responsibilities. Real deployments can use separate stores or logical schemas. Do not create extra copies merely to reproduce the diagram.

Preserve evidence without publishing every record.

Raw fidelity and strong controls are compatible. Minimal business transformation does not mean minimal protection.

RESPONSIBILITY	INVOICE EXAMPLE AND FAILURE RESPONSE
Intake	Bind source, extraction position, source/arrival timestamps and batch or event ID. Validate the delivery envelope and source rights. Reject an untrusted delivery before ordinary processing.
Bronze	Preserve a replayable source representation with controlled access, encryption and justified retention. Capture schema drift and duplicate event IDs. Replay must not create duplicate facts.
Silver	Resolve approved types and business keys; normalize dates; validate currency and entity references. Separate invalid or ambiguous records for owner review. Keep rejection reasons and source links.
Correction	Retain valid time and recorded time. A late payment can change what is now known without silently rewriting what a prior report knew at its cutoff.
Reconciliation	Compare counts, totals and critical records with the authoritative sources. Declare tolerances and exclusions; a syntactically valid amount can still be wrong.

An unknown currency code is quarantined rather than guessed. An exact repeated event is handled by the deduplication rule. A revised event with the same business key is processed through the correction/version rule, not discarded as if it were an identical duplicate.

Decide whether a rejected row permits partial publication. A reporting deadline, materiality threshold or required total may demand a hold; another use may permit a clearly bounded partial result. Record that behavior in the product contract.

PRODUCT CONSTRUCTION / PUBLICATION

Make the interface a maintained commitment.

A validated dataset becomes a supported product when a team publishes and maintains a consumer contract.

PUBLICATION DECISION	EVIDENCE FOR THE SYNTHETIC INVOICE LINES PRODUCT
Meaning and grain	One row per invoice line and declared version; stable invoice/line/entity identifiers; amounts in stated currency; defined tax and credit treatment.
Authority and scope	Named source authority and product owner; supported consumers and purposes; unsupported decisions are explicit.
Serving and access	Versioned table/view or API, examples and entitlement checks. All supported routes enforce the same applicable rights; an unrestricted alternate extract is not a workaround.
Quality and freshness	Reconciliation, source completeness, measured delay, cutoff and final/preliminary status; define the response when a requirement fails.
Operation and change	Support contact, monitoring, incident runbook, recovery, compatibility rules and consumer migration notice.

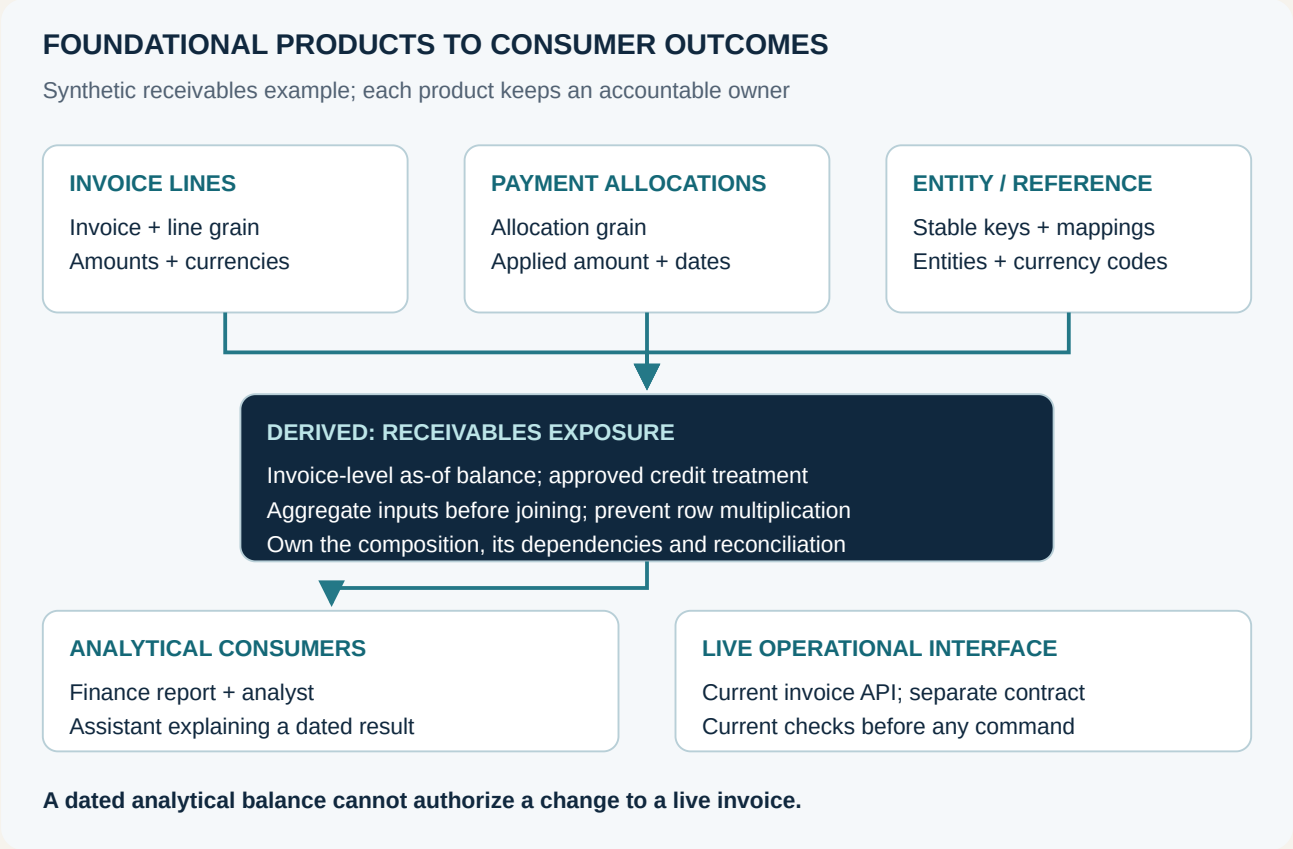
Gold need not be an aggregate. Invoice Lines remains detailed so approved consumers can compose it accurately. A different organization may publish a supported product from a silver representation; the contract and operating responsibility matter more than the color label. Unpublished engineering tables remain internal in this reference design.

Catalog registration makes the product discoverable; it does not certify its correctness or grant access. Certification is scoped to the product version and supported use. See the Data Product Certification Standard for the evidence and promotion decision.

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.

Compose products without losing accountability.

Foundational products provide reusable capabilities. A derived product owns the new meaning created by composition.



RJL product relationship map. The live interface is a separate serving path, not a current-state guarantee inherited from the analytical snapshot.

The Receivables Exposure owner agrees the invoice-level calculation, period, entity scope and credit treatment with Finance. Source-aligned owners retain responsibility for their own interfaces. The derived owner manages compatibility, critical dependencies, permissions and reconciliation of the combined result.

Foundation does not mean untouched source data. Derived does not necessarily mean several domains, aggregation or greater transformation complexity. Describe the actual inputs, grain and consumer promise.

PRODUCT CONSTRUCTION / DELIVERY CHOICES

Choose the path by the use, not the label.

Products can be analytical, operational or knowledge-based, and foundational or derived within each type.

TWO DIMENSIONS; DIFFERENT DESIGN DECISIONS

Delivery type is separate from foundational or derived relationship

	FOUNDATIONAL / SOURCE-ALIGNED	DERIVED / PURPOSE-ALIGNED
ANALYTICAL	**EXAMPLE** Invoice lines under contract Detail + history + corrections	**EXAMPLE** Receivables as-of snapshot Defined joins + reconciliation
OPERATIONAL	**EXAMPLE** Current invoice read API State + latency + failure contract	**EXAMPLE** Computed collections signal Fresh inputs + explicit authority
KNOWLEDGE	**EXAMPLE** Approved policy collection Versions + source permissions	**EXAMPLE** Corpus for invoice explanations Selection + expiry + citations

Foundational is not raw; derived need not cross domains or aggregate records.

RJL implementation taxonomy. The broad operational and knowledge product terminology is an extension of the analytical data-mesh convention.

Analytical pipelines can use streaming updates. Operational feeds can have bounded delays. Set latency, consistency and recovery expectations for the actual decision rather than assuming all operational products are instantaneous.

Knowledge products require source permissions, parsing or OCR where needed, versioned chunks, a supported index and tested permission/deletion propagation. An embedding or knowledge graph alone is not a product contract.

PRODUCT CONSTRUCTION / WORKED COMPOSITION

Protect the grain before calculating the balance.

Synthetic invoice INV-100 has one currency and one legal entity. At cutoff T0 its gross lines total 1,000, approved credits total 100 and applied payments total 400.

STEP	CALCULATION OR CHECK
Declare the definition	Outstanding = gross invoiced amount minus approved credits minus applied payments, for the same invoice, entity, currency and as-of cutoff. Other adjustment types require explicit rules.
Preserve the input grains	Invoice Lines has line grain; Payment Allocations has allocation grain; credits can have their own grain. Do not join every detail row and then sum multiplied amounts.
Aggregate before joining	Aggregate each input to the declared invoice/entity/currency/cutoff key. Join those reconciled results using approved cardinality and missing-record behavior.
Calculate the example	1,000 - 100 - 400 = 500 outstanding at T0. This simplified arithmetic assumes all relevant adjustments are represented; it is not a general accounting standard.
Handle ambiguity	Reject or flag an unresolved entity, unknown currency or unmatched allocation according to the contract. No assumed zero or automatic FX conversion hides the problem.
Reconcile and serve	Compare with the authoritative balance at a consistent cutoff. Publish source/version references and explain any expected timing differences.

An invoice with two lines and two payment allocations is a useful negative test: a raw many-to-many join can multiply both totals. Include that case, duplicate events, missing credits and a backdated payment in the release evaluation.

The assistant may explain the dated result with citations. If asked to change the live invoice, it must obtain current state and follow the separately authorized command path.

PRODUCT CONSTRUCTION / CONSUMER COMMITMENTS

State which clock and which failure matter.

A daily Finance report and an invoice assistant can consume related data while needing different service commitments.

CONSUMER	COMMITMENT TO DEFINE	BREACH RESPONSE
Daily Finance report	Required sources, reporting cutoff, currency and finality; measure delivery after the agreed source-complete condition.	Hold final publication if reconciliation fails. Permit a provisional version only under its approved terms.
Dated-balance assistant	Source cutoff and version returned with the answer; maximum permitted age and supporting document rights.	Disclose the dated scope; stop or use an approved current source when the requested use exceeds it.
Current invoice application	Measured source-to-serving lag, query latency distribution, availability and consistency requirements.	Do not substitute a prior-night snapshot for a prohibited stale operational response.

An SLI measures behavior, such as the proportion of eligible daily publications delivered by their agreed deadline. An SLO is the target for that measure over a specified window. An SLA states the agreement and consequences of missing it. Not every internal target is an SLA.

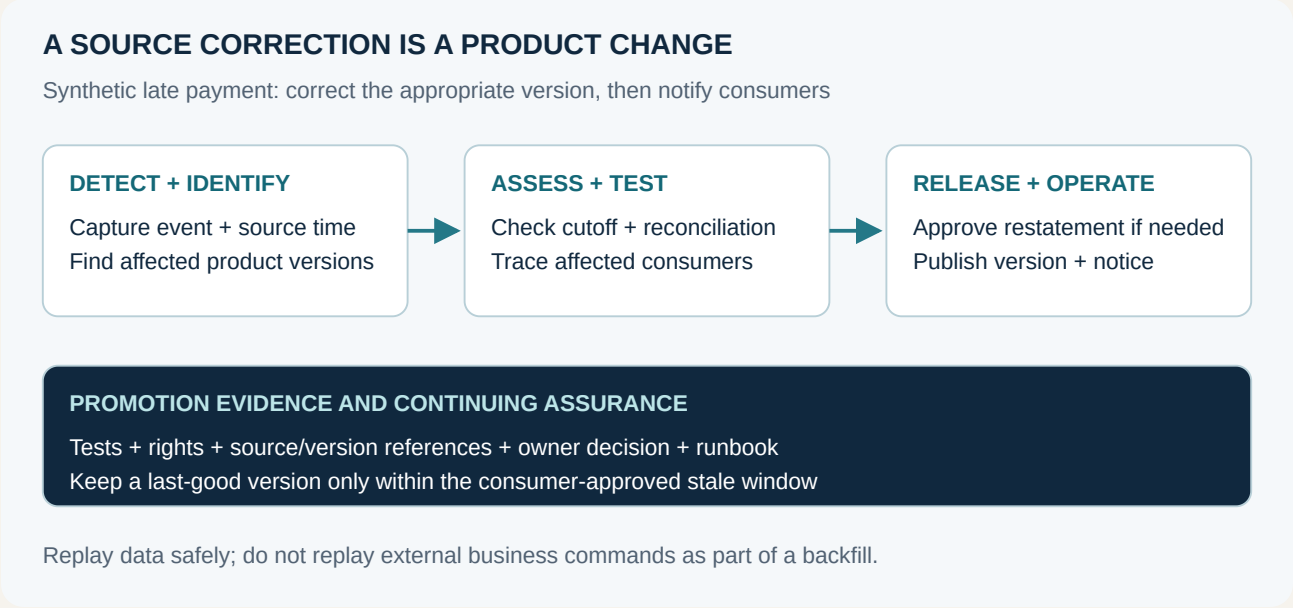
Define the denominator, timezone, source dependency, exclusions, evaluation window and accountable operator. Monitor end-to-end freshness as well as the successful pipeline run: a pipeline can run on time while its input is incomplete or old.

Numerical targets need workload, source, capacity and cost evidence. Agree them with the consumer; do not turn an illustrative seconds-level feed into a universal promise.

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.

Keep the promise after the first publication.

At T1 a further payment of 200 arrives. Under the simplified example definition, the current balance becomes 300; the published T0 record remains a dated version.



RJL correction and promotion loop. A closed-period restatement follows its own authorized process; it is not a silent overwrite.

Determine event time, arrival time and affected cutoffs. Recompute eligible versions, reconcile and notify affected consumers. Preserve the prior evidence required for historical interpretation. A source-field change also needs contract impact and compatibility tests before publication.

When a second consumer requests access, compare purpose, grain, rights, latency and service cost before reusing the interface. Reuse the existing contract where it fits; publish a derived interface or justified separate product where it does not.

Review adoption, accepted outcomes, support effort, incidents and full cost. Expand, improve, consolidate or retire based on evidence. A permanent catalog badge and a growing consumer count are not sufficient grounds for continued investment.

07 / FRESHNESS AND FINALITY

State when the answer is fit to use.

Freshness means more than a refresh frequency. It covers the date represented, completeness and whether the result can still change.

COMMITMENT	CONTRACT MUST SPECIFY	FAILURE RESPONSE
Current	Event or state delay; measurement point; availability and ordering expectations.	Restrict or stop time-sensitive actions; use an approved fallback.
Daily	Business cutoff, source completeness and delivery deadline.	Mark delayed inputs; notify consumers; apply a defined stale-data window.
Period-end	Valuation period, final/preliminary status, restatement and reconciliation rules.	Block final reporting if required evidence is missing; preserve approved versions.

An hourly refresh can still contain yesterday's incomplete source. A quarter-end reserve figure can be fresh and still preliminary. Make these distinctions visible in the interface and catalog. A PDF service description isn't enough.

Agree which clock starts the service measurement: source event time, source availability or publication time. Describe timezone, late-arriving records, duplicate handling and replay. A live event feed also needs idempotent consumers and a recovery process.

Keep valid time apart from recorded time. Valid time is when a fact was true in the business, such as the date a policy change took effect. Recorded time is when the platform learned it. A backdated correction changes the first without rewriting the second. So a product that keeps both can answer two questions: what was true on a date, and what was known on that date.

Define a maximum stale period by use. Serving the last good version can suit a historical report and be wrong for a time-sensitive action. A warning doesn't authorize continued use outside the contract.

ILLUSTRATIVE SERVICE LEVELS

A team might propose a daily close deadline or a seconds-level status feed. Treat these as design targets to validate. They aren't default commitments, and they aren't evidence of achieved performance.

Choose the shared responsibility deliberately.

Identifiers, hierarchies and codes need coherent stewardship. Not all of them need centralized authoring.

MDM STYLE	USEFUL WHEN	RESPONSIBILITY TO ESTABLISH
Registry	Source systems remain authoritative; matching links their records.	Identity matching, linkage quality and resolution of conflicts.
Consolidation	A harmonized view is needed for reporting or analysis.	Survivorship, provenance and limitations of the consolidated view.
Coexistence	Master changes move between central and source systems.	Synchronization, conflict handling and which changes each system may make.
Centralized authoring	A shared master is the authoritative place for specified changes.	Stewardship workflow, access, distribution and recovery.

DAMA-DMBOK discusses these established styles [3]. Choose by authority, operational needs and migration constraints. A registry won't magically resolve conflicting business definitions. And central authoring can become a bottleneck if its service model is weak.

Treat reference products as products: a named owner, controlled codes, effective dates, supported distribution and change notice. Domain consumers stay responsible for interpreting them appropriately. Record who owns enterprise standards, and who can approve domain-specific extensions.

For licensed third-party data, put permitted purposes, onward sharing, retention and deletion obligations in the contract. A copy or derived product must never silently discard the original use restrictions.

SEPARATE IDENTITY FROM MEANING An enterprise party identifier can connect records. It doesn't establish that every domain defines customer, exposure or active status the same way.

09 / THE PRODUCT CONTRACT

Write the promise before publication.

A compact contract connects business meaning to engineering tests and operational responsibility.

CONTRACT ELEMENT	MINIMUM USEFUL CONTENT
Purpose and consumers	Supported decisions and uses; prohibited or unsupported uses.
Meaning and grain	Definitions, units, dates, keys, joins and final/preliminary status.
Interface and version	Schema or API, examples, compatibility and change notice.
Freshness and availability	Measurement point, thresholds, stale-data policy and fallback.
Quality and evidence	Critical tests, tolerance, reconciliation and source provenance.
Classification and rights	Sensitivity, access and purpose restrictions, retention and licensing.
Owners and operation	Accountable owner, delivery/run contacts, incidents and escalation.
Lifecycle and retirement	Current status, dependencies, migration plan and notice period.

Version the contract with the product, and reflect it in machine-readable configuration where practical. Automated tests and runtime controls enforce the supported parts of the promise. A human-readable permitted-use statement on its own enforces nothing.

Review the contract with a representative consumer. Ask them to make an authorized query or API call, interpret the output and handle an unavailable or stale response. That exercise usually exposes missing information faster than another metadata form.

RELEASE EVIDENCE — Keep the approved contract, test results, source/version references and decision record. An updated contract can require consumer migration or risk review even when the physical schema hasn’t changed.

Test the data needed for the decision.

Quality depends on purpose. Define a failure and a response for each critical element.

QUALITY DIMENSION	ILLUSTRATIVE TEST
Completeness	Required claim-payment fields are present; expected source batches have arrived.
Uniqueness	A payment identifier appears once within the agreed business key and version.
Timeliness	Source and publication delay remain within the product commitment.
Validity	Currency, status and date values conform to approved domains and rules.
Accuracy	Paid amounts reconcile to the authoritative payments record within an approved tolerance.
Consistency	Coverage and loss dates align with policy terms; cross-product totals reconcile for the same purpose.

These dimensions are established data-management concepts [3]. A valid-looking value isn't necessarily accurate. Material accuracy claims need reconciliation to an authoritative record, or another justified evidence method.

Don't reject a claim just because the policy is currently closed. A valid claim can be reported or settled later. Coverage checks need the relevant policy terms and event dates. Backdated changes and restated balances need similar care.

Test critical data elements first. Record thresholds, exceptions, affected consumers and escalation. Quarantining a defective record is different from stopping a whole product. The right response depends on the materiality and permitted use of the affected output.

MEASURE CONSUMER IMPACT A failed test matters because it threatens a use. Link quality incidents to the decisions, workflows and deadlines they affect, then prioritize the repair accordingly.

Certification is a release decision.

A lifecycle describes where work stands. Certification describes the evidence supporting a particular use.

LIFECYCLE STATE	REQUIRED EVIDENCE OR ACTION
Proposed	Named need, likely consumers, owner and value hypothesis.
Designed	Boundary, contract, architecture and applicable control requirements.
Building	Versioned implementation, tests and consumer feedback.
In review	Release evidence, risk assessment and operational readiness.
Published	Supported consumption at an explicitly stated assurance level.
Retired	Dependency migration, access withdrawal, retention and cleanup completed.

Publishing a product doesn't certify it for every purpose. Record the certification scope, approver, product version, evidence and review trigger. Exploratory use and material production decisions may need different assurance.

These delivery stages sit alongside the certification states in the Data Product Certification Standard: sandbox, managed sandbox, certified, degraded, deprecated and retired.

Changes to definitions, source quality, allowed use or serving controls can invalidate earlier evidence. Monitor commitments after release. When a material failure happens, the status should show it; an unconditional trust badge left in the catalog hides it.

Before a production release

- Owner and consumer approve a usable contract and supported interface.
- Critical tests and reconciliation pass; essential lineage reaches the source.
- Permitted and prohibited access paths are tested.
- Monitoring, incident ownership, fallback and recovery are ready.
- Consumers understand limitations, change policy and escalation.

Keep certification criteria consistent with the organization's adopted standard. This checklist is an implementation aid. It doesn't replace that standard.

Use DCAM v3 as a reference, not a label.

A product portfolio depends on program capabilities as well as individual product quality.

DCAM V3 AREA, SHORTENED	RJL PRODUCT-PROGRAM QUESTION
Strategy and business case	Are product priorities linked to funded business outcomes?
Program and funding	Can teams maintain and operate the products they publish?
Architecture	Are business, data and technology designs coherent?
Business Data Knowledge	Can consumers interpret definitions and permitted use?
Data quality	Are critical issues measured and resolved with evidence?
Governance	Are decision rights and exception routes effective?
Operations, risk and control	Are commitments monitored and controls tested?
Analytics management	Are downstream models, analytics and agents governed?

DCAM v3 consolidates architecture into one component and adds Business Data Knowledge [4]. I've shortened the area labels above for readability. The questions are an RJL adaptation. They aren't official DCAM capability statements, scoring criteria or an authorized assessment.

For internal prioritization, use evidence such as a current contract, an access test, an incident record or a reconciled result. Pick the next improvements by business exposure and delivery constraint. The weakest numerical score isn't automatically the most important investment.

CONNECT PRODUCT AND PROGRAM EVIDENCE A product can be well built while its ownership, funding or cross-domain governance stays fragile. Review both levels before extending the portfolio.

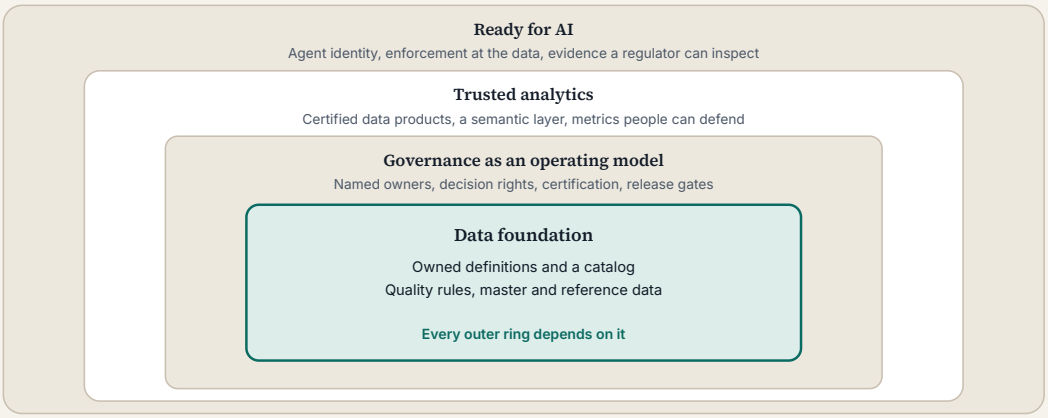


Exhibit: Governance is built from the center out. The outer ring cannot be bought on its own.

Track use, reliability and economics.

Consumer counts alone cannot show whether a product deserves further investment.

MEASURE	DEFINITION AND DECISION
Active consumption	People, applications or agents using the supported interface in a stated period; distinguish types.
Adoption and reuse	Eligible-consumer adoption and additional supported uses; review low reuse in context.
Time to first use	Elapsed time from a valid request to a working, authorized consumption path.
Commitment breaches	Missed freshness, quality or availability promises and their downstream impact.
Change burden	Consumer migrations, breaking changes, support requests and unresolved issues.
Cost and value	Build/run cost and change in the named workflow against an agreed baseline.
Consumer experience	Satisfaction and support friction by consumer group, with a clear denominator.

A product with one critical application consumer can still deliver material value. A product with many consumers can still be unreliable or uneconomic. Before deciding to expand or retire it, review the outcome, operating risk, alternatives and lifecycle.

At 30, 60 and 90 days after release, ask three questions. Can eligible consumers use it without bespoke extracts? Do the commitments hold? Is the support effort sustainable? These are review points, not automatic pass/fail rules.

MAKE BENEFIT CLAIMS REPRODUCIBLE	State the observation window, baseline, affected population and full operating cost. Never turn a conceptual reuse argument into an unmeasured savings claim.

14 / ILLUSTRATIVE PORTFOLIO

An insurer publishes a coherent set.

A portfolio can support examiner work and analytical oversight without merging every asset into one policy file.

PRODUCT / ACCOUNTABLE OWNER	DELIVERY AND ROLE	COMMITMENT TO AGREE
Claim payments / claims owner	Source-aligned analytical history; payment reconciliation.	Daily completeness, history and restatement rules.
Claim status / claims owner	Source-aligned operational API; current case state.	Maximum source-to-consumer delay and fallback.
Claim cost to close / named business owner	Aggregate analytical measure; finance approves reconciliation.	Definition, finality, allocated-cost rules and period cutoff.
Examiner case summary / examiner-workflow owner	Consumer-aligned output using approved data and knowledge.	Case scope, citations, evaluation and human verification.
Procedure knowledge / procedure owner	Knowledge content used by permitted retrieval; a separate content dimension.	Validity dates, current access rights and revocation propagation.

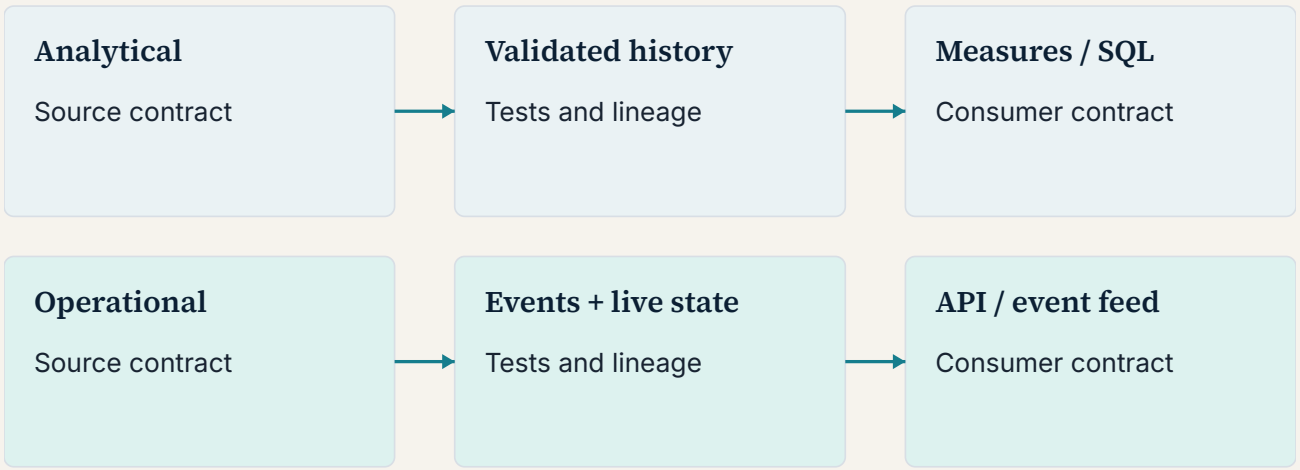
The payment product gets reconciled before it supports the cost measure. The live status API uses an operational serving path instead of waiting for a daily analytical refresh. The summary tool obtains definitions and document context before retrieval, and cites versions in its output.

If a source is late, each consumer follows the relevant product policy. A historical report might use a marked last-good version within a defined window. A time-sensitive action may stop. The case summary should explain missing evidence. Silently filling the gap is never acceptable.

ILLUSTRATIVE DESIGN The portfolio and commitments are proposed choices. They don’t describe a client deployment, an achieved service level or a regulatory approval.

Make the first release repeatable.

A first product should leave behind a usable asset and a delivery pattern another team can understand.



Two serving paths under common ownership, contracts and controls.

FIRST-QUARTER FOCUS	EVIDENCE BEFORE EXPANSION
Define	One outcome, owner, consumer, product boundary and critical definitions.
Build and prove	Working interface, representative consumer use and tested minimum controls.
Run and learn	Service monitoring, incident ownership, adoption and cost baseline.
Prepare reuse	Documented pattern, dependency map, support capacity and next justified consumer.

Timing depends on data condition, access, available capacity and procurement. A 90-day plan is a scoped planning aid, not a universal delivery promise. Limit the initial use, and make the exit evidence explicit.

THE TEST OF SUCCESS A consumer can interpret and use the product, the team can operate it, and an executive can explain its value and exposure.

Sources and scope.

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

[1] *Data Mesh*. Zhamak Dehghani, Data Mesh: Delivering Data-Driven Value at Scale, 2022; and her public principles article, December 2020. Product terminology here is adapted for implementation. martinfowler.com / data mesh principles

[2] *Bounded Context*. Martin Fowler, explaining the domain-driven design concept introduced by Eric Evans. Purpose-specific meanings need explicit mappings. martinfowler.com / bounded context

[3] *DAMA-DMBOK*. DAMA International, Data Management Body of Knowledge, second edition. Referenced for data-management disciplines, MDM styles and quality dimensions. dama.org / DMBOK

[4] *DCAM v3*. EDM Council / EDM Association. Architecture is consolidated and Business Data Knowledge added. Shortened labels and questions here are an RJL adaptation. edmcouncil.org / DCAM

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	Ownership, teams, decision rights and funding.
Data Products in Practice (this paper)	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 dataset that consumers struggle to use. RJL Solutions helps define the product boundary, accountable owner, contract, delivery path and evidence needed for reliable consumption.

OPERATING MODEL **The Data and AI Operating Model**
Ownership, teams, decision rights and funding.

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.