

ORACLE

Oracle GoldenGate Data Fabric

From Committed Change to Trusted Action: Capture once. Govern continuously. Activate everywhere.





Executive Summary

Every important AI answer, alert, and automated action is downstream of a committed business event. The value of that event decays while data waits in disconnected pipelines.

Enterprises have invested heavily in AI, yet many initiatives still struggle to reach measurable production value. Gartner identifies siloed data, fragmented solutions, overlapping capabilities, and an inability to provide AI-ready data as recurring constraints [1]. Beneath the AI model sits a more mundane obstacle: operational context arrives late, incomplete, hard to verify, or fragmented across pipelines. The hardest enterprise data problem is turning authoritative Systems of Record (SoR) business changes into current, governed, reusable data products before their value is lost.

One real-time data fabric, three outcomes:

OUTCOME 01

Operate

Keep mission-critical systems synchronized and resilient across planned and unplanned change.

OUTCOME 02

Activate

Deliver governed, current data products to applications, analytics, agents, and AI across distributed data centers and clouds.

OUTCOME 03

Optimize

Preserve platform choice and place workloads where performance, locality, sovereignty, and economics work best.

Oracle GoldenGate Data Fabric captures committed change once, preserves transaction context, and continuously turns it into governed streams, tables, APIs, and data products across the operational, analytical, and AI platforms the enterprise has chosen. Built on GoldenGate's mission-critical change-data foundation, it connects how the business operates with how the enterprise analyzes, decides, and acts, without requiring every workload, DB, or lakehouse to move onto one platform [2], [3].



Why GoldenGate: more than thirty years keeping mission-critical systems synchronized, migrated, and recoverable; heterogeneous and multicloud integration across supported databases, streams, and lakehouses; Veridata verification that proves data arrived correctly; and managed deployments across the most common environments [4]–[9].

Three critical technical advantages:

- (1) a shorter path from committed data events to trusted actions, measured as commit-to-action latency
- (2) one capture feeding many products, with stronger verification and recovery evidence
- (3) platform choice preserved: across five managed platforms, the highest modeled cost for equivalent Spark pipelines ran about 54 percent above the lowest [18]

Where to start: baseline one journey's commit-to-action latency, start a trial of GoldenGate Data Fabric, and measure the result (Chapter 10).

What's new in GoldenGate Data Fabric

A unified, fully managed, AI-assisted product experience turns GoldenGate's real-time foundation into governed data products that can be developed, operated, and consumed across distributed environments.

UNIFIED EXPERIENCE

Operational and analytical data fabrics

Unifies design and operations across replication, data streams, ZeroETL ingestion, ETL, E-LT, continuous transformation and loading, reverse ETL, data preparation, and governed publication. The experience serves GoldenGate operators, forward-deployed data engineers, application developers, data architects, and data product managers.

AI-ASSISTED DEVELOPMENT

Multiple levels of control

Users can create pipelines through natural-language prompting, low-code editors, or a high-code domain-specific API. AI assists with pipeline construction, mapping, and transformation, while individual pipeline steps can use supported external models. GoldenGate Data Fabric is not itself a model-training or model-hosting platform.

GOVERNED DATA PRODUCTS

Built from operational change

Publishes current data as governed tables, Datasets, streams, APIs, and Gold-layer serving structures. Policies, security controls, validation, profiling, quality checks, verification, compare/repair, metadata, and lineage evidence are associated with the product lifecycle.

OPEN EXECUTION

Portable by design

Logical pipelines and data products are decoupled from a single physical platform. Customers can use supported database and Spark execution choices, relational and open lakehouse table formats, Oracle AI Data Catalog, compatible Iceberg REST catalogs, and customer-selected infrastructure.

PRODUCTIZED DELIVERY

Self-service at enterprise scale

Reusable Datasets, pipeline patterns, metadata, guardrails, branching, merging, deployment controls, fabric visualization, telemetry, and estate management turn implementation knowledge into repeatable operating practice. More teams can deliver and operate pipelines without making every use case dependent on a new embedded expert engagement.



Table of Contents

PART I — The Missing Layer.....	6
Chapter 1. Every AI Decision Starts from Operational Events	6
Chapter 2. Platform Convergence Still Leaves an Operational Gap	8
Chapter 3. New Mission-Critical Data Fabric Experience	10
 PART II — The GoldenGate Difference	16
Chapter 4. One Change, Many Products	16
Chapter 5. Trust That Travels With the Flow.....	19
Chapter 6. Choice Without Chaos: Portable, Open, and Sovereign Data	21
 PART III — Putting GoldenGate Data Fabric to Work	25
Chapter 7. From Design to a Governed Product.....	25
Chapter 8. Fit the Enterprise You Already Have.....	30
 PART IV — Journeys, Economics, and Proof.....	33
Chapter 9. A Lighthouse Journey	33
Chapter 10. Economics and Measurable Outcomes.....	35
Chapter 11. Your Next Step.....	37
 Appendices	38



PART I — The Missing Layer

Chapter 1. Every AI Decision Starts from Operational Events

At 9:41 on a Tuesday morning, a customer taps "Place Order." An operational database commits a transaction that, in one atomic act, creates the order record, decrements inventory, and flags the item as awaiting shipment. At that instant — and not before — the order is real. Everything the enterprise will ever know, decide, or automate about that order is downstream of that commit.

The enterprise runs on committed state changes. Orders, payments, inventory movements, account updates, claims, and shipments all become authoritative inside operational systems, at a transaction boundary, in a specific order.

1.1 The Freshness Gap Between Operations and AI

Now follow that order into a typical modern estate. The order record arrives in the analytics platform on the next streaming micro-batch. The inventory decrement arrives fifteen minutes later in a scheduled ETL job. The shipment flag arrives the following morning. Three changes born as one atomic business event have been scattered across three pipelines on three schedules.

An AI agent assigned to manage fulfillment checks its three systems at 9:50 that morning. Order: confirmed. Inventory: available, because the decrement hasn't landed yet. It approves an outbound shipment for stock committed to someone else nine minutes earlier. Nothing was corrupted and every query returned. The failure was one of transactional completeness: the agent saw a partial view of a business event and acted as though it were complete.

AI does not fix stale or partial data. It scales the unintended consequences: a wrong dashboard produces a wrong chart, while a wrong AI agent executes a wrong business process at machine speed, before anyone notices.

For decades a fifteen-minute ETL lag was an acceptable trade-off because a human applied judgment before acting. As AI systems move from analysis to action, that checkpoint disappears for a growing class of decisions, and with it the tolerance for stale, partial data.

1.2 Why Better Models Do Not Repair Stale Inputs

Trusted AI depends in part on the freshness, provenance, quality, semantics, and policy posture of the data supplied to the model. A retrieval-augmented application grounding its answers in yesterday's batch extract confidently answers questions about a world that no longer exists. Model quality determines how well the system reasons; the data path determines what world it reasons about.

1.3 The Cost of Too Many Hops

How did we get here? One hop at a time: database, CDC connector, message bus, object store, transformation engine, lakehouse, semantic layer, agent. Each hop was a reasonable local decision. Together they levy a hop tax: accumulated lag, duplicated transformation logic, more failure and recovery surfaces, and ownership gaps where nobody can say who owns the wrong number on the dashboard. Practitioners feel this daily: monitoring lag, working out why a pipeline is stuck, re-running backfills, and paying for idle capacity across four platforms that each own one hop.

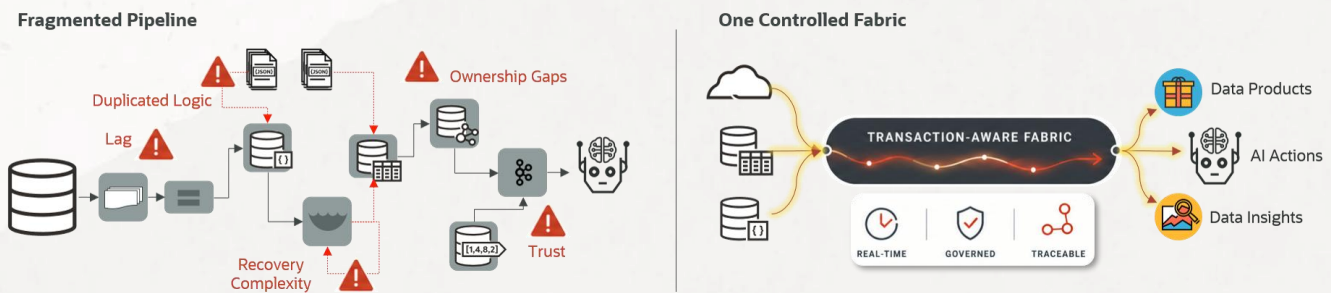


Fig. 1.1. The Hop Tax.

1.4 Commit-to-Action Latency: the Key Metric

Vague claims about being "real time" help nobody. This paper applies a measurable alternative: commit-to-action latency, the elapsed time from a system-of-record commit to a trusted business response, decomposed into six stages.

1. Commit-to-capture — from the source commit to the change entering the fabric.
2. Capture-to-publish — transformation, validation, and publication into a governed product.
3. Publish-to-consume — a consumer (application, analyst, agent) reads the current product.
4. Consume-to-decision — a human or model reaches a decision on the data.
5. Decision-to-action — the decision changes something in the business.
6. Action-to-observed-result — the action commits new operational state, closing the loop.

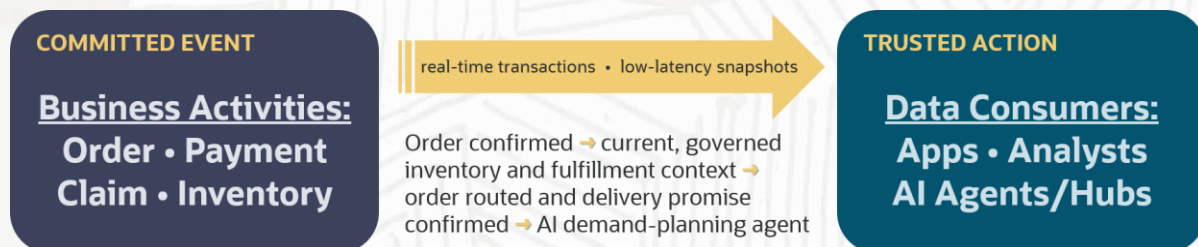


Fig. 1.2. Elapsed time from System of Record (SoR) commit event to a trusted business response.

GoldenGate Data Fabric directly governs the early stages and provides the current, traceable data the later stages require. The framework turns "we need real-time data" into a baseline you can measure this quarter and a target you can pilot against in ninety days.

Chapter 2. Platform Convergence Still Leaves an Operational Gap

Modern centralized data management platforms are genuinely good. Unified storage, engineering, governance, semantics, analytics, and AI development inside one environment is a real simplification, and enterprises are right to standardize on them [11]. This paper argues something complementary: even the best data platform only optimizes what happens inside its own boundary, and **the enterprise does not live inside one boundary.**

2.1 Six Archetypes, One Converged Promise

Across the market, vendor narratives have converged on the same vocabulary: unified data and AI, metadata and governance, data products, open formats, agentic experiences, real time. Rather than a feature-by-feature battle, it is more useful to classify the market by architectural center of gravity; coexistence patterns follow in Chapter 8.4. Gartner evaluates these platforms through four use cases and the CODE dimensions of capability trajectory, operating model, deployment focus, and ecosystem strategy [1], [10].

Archetype	Center of gravity	GoldenGate relationship
Platform-centered suite	One integrated data, analytics, and AI environment	Complement the platform with current, trusted operational change from across the estate
Context / governance-centered fabric	Catalog, semantics, policy, quality, lineage	Feed the context layer with transaction-aware, continuously governed flows
Logical / federated fabric	Access data without physically centralizing it	Combine logical access with durable, event-driven movement where the use case requires it
Ontology-to-action platform	Ontology, applications, decisions, agents	Supply authoritative, fresh operational events and products within an open data-platform architecture
Entity-centric product platform	A complete governed view of an entity	Produce entity-ready streams, tables, and APIs from heterogeneous committed change
Integration and quality suite	Connectors, ETL, CDC, quality	Differentiate on operational heritage, transaction context, HA, verification, and multicloud deployment

GoldenGate Data Fabric's own center of gravity is committed operational change, which puts it closest to the integration suites and beside the other five. Decades of effort to consolidate onto a single enterprise data warehouse taught the industry that total centralization rarely happens and is often not desirable. The data fabric paradigm points to the rational alternative: **standardize the fabric, not the platform.** Let each platform evolve to serve its domains and use cases, but do not force whole companies or business units into one walled garden. Every platform carries its own data gravity, and



an IT estate will always have several centers of it. The fabric is the governance and data-product layer that spans them, from operational systems to downstream BI and AI platforms, and turns platform choice into economic leverage.

A distinct fabric layer across the enterprise

Connect multiple platforms. Deliver governed data products directly to applications and AI.

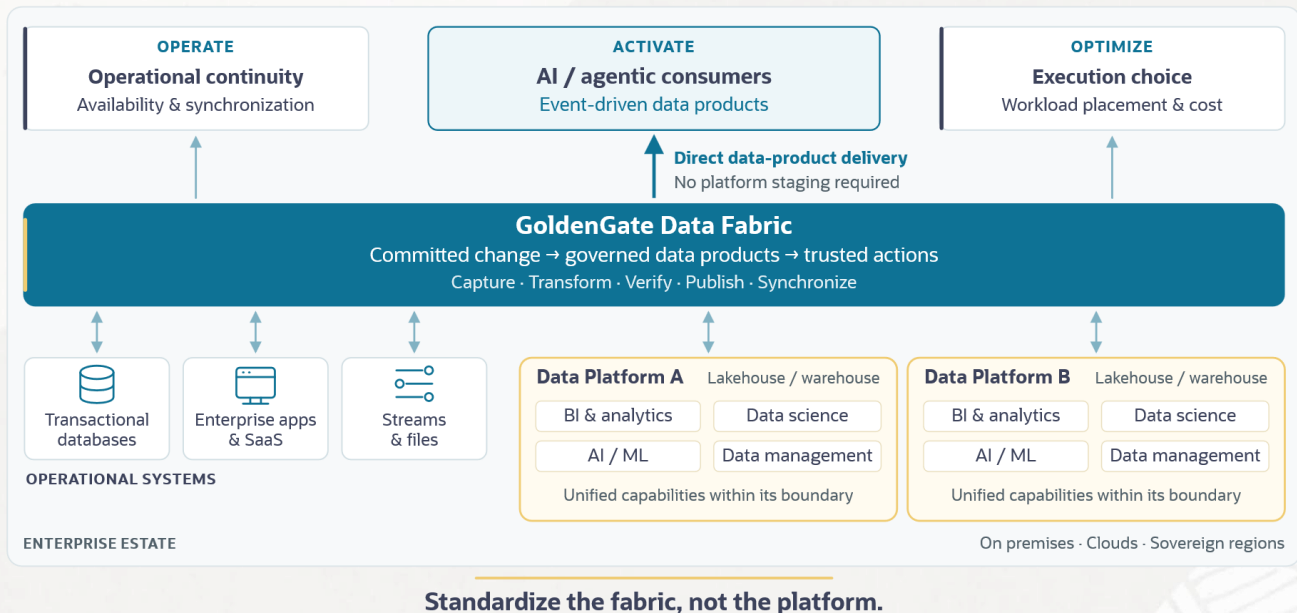


Fig. 2.1. Platform boundary vs. enterprise estate – a complete data fabric should span operational use cases too.

2.2 The Gap Between Platform Data and Enterprise Operations

However complete the platform, the enterprise still runs on multiple transactional databases, packaged applications, several clouds, sovereign environments, messaging systems, edge sites, and long-lived legacy systems. Data residency regulation keeps certain data inside jurisdictional lines regardless of architectural preference, and production databases stay on-premises because the applications above them cannot move. This is the durable shape of the modern enterprise, and waiting it out is not a plan.

2.3 Why the Lakehouse Is Not an Operational Database

Open table formats are a real advance: atomic table commits, schema evolution, time travel, multi-engine access. But a table-level commit in a lakehouse is not the same as preserving a source transaction boundary that spanned multiple operational tables. An Iceberg table receiving order and inventory changes through separate pipelines commits each independently, exposing readers to inconsistent cross-table state between cycles. Nor do analytical platforms replace high-concurrency, low-latency operational state or mission-critical synchronization.

The lakehouse is a superb library. It is not a cash register.

Chapter 3. New Mission-Critical Data Fabric Experience

In plain terms: GoldenGate Data Fabric is one fully managed data fabric that spans the mission-critical operational systems that run the business and the analytical and AI platforms that learn from it. It turns committed business events into governed data products, and governed data products into trusted actions, continuously, wherever the business runs.

Four stages describe how data flows through the product:

Committed changes → real-time events → governed products → trusted actions

Committed business events enter the fabric at low latency with transaction context intact. The fabric captures, governs, and verifies them, keeping controls, provenance, and processing evidence linked to the source of truth. It publishes current tables, Datasets, streams, and APIs that are reusable across the enterprise, with semantics managed in data catalogs. Applications, analysts, BI, and AI agents then act on a current, traceable foundation.

Chapter 1 named the metric this shortens: **commit-to-action latency**. Everything in this chapter exists to make that number smaller and verifiable.

3.1 GoldenGate is a Platform

Reading committed changes from a transaction log is where GoldenGate starts. It is not what GoldenGate is. The same platform that captures a change keeps a payment system running through a datacenter failure, moves a bank to a new database with minutes of downtime, feeds fraud models with events seconds old, and lands a lakehouse's Bronze layer without a batch schedule. Three outcomes have organized that work for thirty years, and they organize this launch [4], [19]:

- **Operate.** Keep mission-critical systems synchronized and resilient across planned and unplanned change: high availability, disaster recovery, zero-downtime migration and upgrade, active-active services, and MAA Platinum architectures [16].
- **Activate.** Deliver governed, current data to the places that act on it: application data streams, warehouses and lakehouses, vector and search indexes, and the APIs and grounding products that applications and agents read.
- **Optimize.** Preserve platform choice and place workloads where performance, locality, sovereignty, and economics work best, across on-premises, edge, OCI, dedicated and sovereign regions, and partner clouds.

The engineering beneath those outcomes is what makes them safe to bet on. GoldenGate captures committed changes from transaction logs rather than querying source tables, and it is designed to preserve commit order and transaction boundaries and to support recoverable, exactly-once delivery in supported, correctly configured replication patterns [4], [5]. For Oracle databases, capture uses private Oracle-internal APIs not exposed through public LogMiner or XStream interfaces, which is why Oracle Database Engineering and GoldenGate product management confirm it as Oracle's

strongest option for capturing change from Oracle databases [4], [5], [15]. Microservices Architecture provides browser, command-line, and REST interfaces, and OCI GoldenGate uses SSL by default with keys and secrets in wallets and vaults [12], [13]. Capture, delivery, DDL behavior, and target semantics vary by source, target, version, and topology [14].

That is the difference between a CDC connector and a platform. A connector moves rows. A platform keeps the business running while it moves them, proves they arrived, and keeps doing so when the network, the schema, or the target changes.

GoldenGate as a trusted platform:

- ✓ More than 8,000 GoldenGate customers globally; approximately 80% of the Fortune 500 run GoldenGate [4], [15]
- ✓ Attributed customer deployments capturing more than 4 TB of Oracle changes per hour and exceeding one million transactions per second [4], [15]
- ✓ Validated performance above 2.5 million changed rows per second using Parallel Replicat on a quarter-rack Exadata X11M [5]
- ✓ Typical source-commit-to-target-apply latency from milliseconds to single-digit seconds in well-tuned deployments [4], [5], [15]
- ✓ OCI GoldenGate is a fully managed service that runs in OCI and partner-cloud regions, as well as sovereign and secure regions [8], [9]

3.2 A Brief Evolution

Each stage of GoldenGate's history added something new the enterprise could do with a committed change, and none retired the stage before it. The first stage made operational systems survivable: replication, high availability, and migration without a freeze window. The second made them connectable: heterogeneous databases, multicloud targets, application data streams, and lakehouse delivery, with Veridata to prove equivalence and managed cloud services to carry the operational burden. Each extension rode the same asset, transaction-aware movement that mission-critical systems could trust [4], [6]–[8].

The third stage, announced here, makes committed change consumable as a product. The unit of value moves from a pipeline that runs to a data product that is owned, governed, current, and reusable. Data Fabric is the next extension of that record: the outcomes are the same three and the trust travels the same way. What changes is who can build on it, how fast, and with how much evidence.

3.3 GoldenGate Data Fabric

Oracle GoldenGate Data Fabric is a managed design-time, runtime, governance, and orchestration experience that continuously turns committed operational change into governed, traceable, AI-ready data products across databases, lakehouses, clouds, and execution engines, without forcing customers to centralize their data or surrender platform choice [2], [3].



The central idea is simple: data products are the outcome; pipelines are the means. Every integration style the enterprise relies on, from a replication topology to a streaming enrichment, a nightly E-LT job, or a reverse-ETL activation, is designed, governed, and observed in one experience, and every flow ends in a governed data product rather than in a table somebody has to go find.

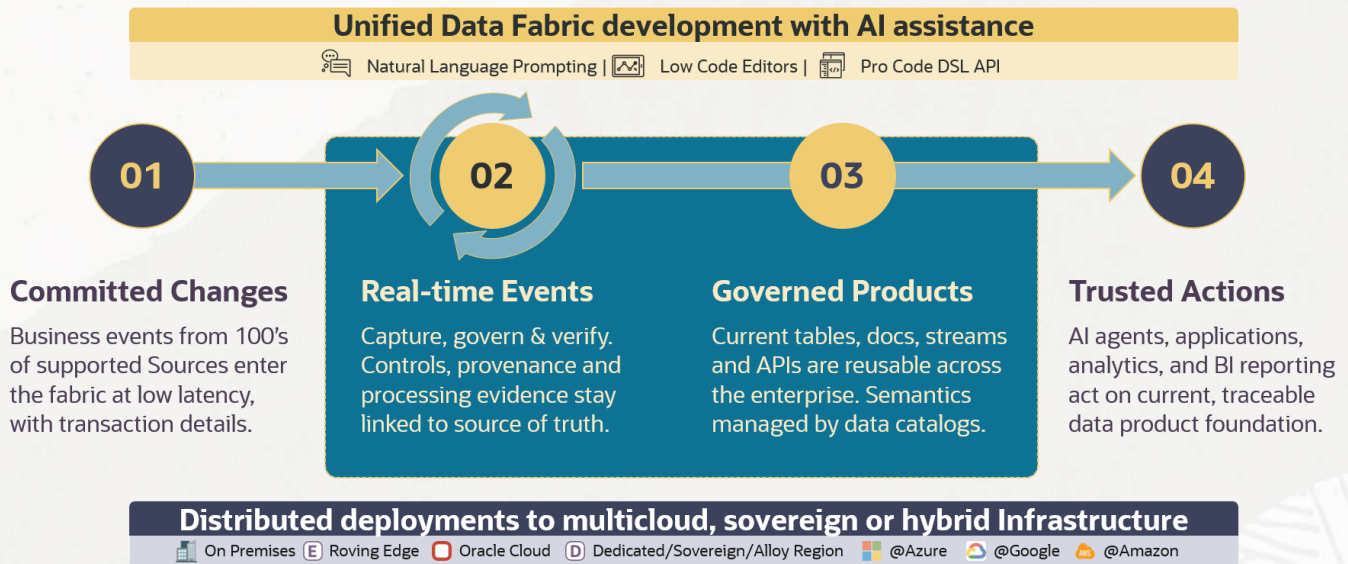


Fig. 3.1. Trusted actions, empowered by a continuous flow of real-time, committed data products.

Its defining product abstraction is the **Dataset**: a durable, logical lifecycle record for a governed data product, carrying transaction and CDC context, schema and endpoint information, pipeline and run evidence, quality and access controls, lineage, and physical-version history. The logical product stays stable while its physical tables are created in customer-selected databases or object stores and registered in the customer's chosen catalogs. That separation of governed identity from any one storage system, catalog, or engine is what gives customers the architectural and economic choice Chapters 6 and 10 quantify.

Three properties distinguish the result from a pipeline tool with a new interface:

- **Trust is established in the flow, not after arrival.** Policy, validation, profiling, quality, provenance, and processing evidence are applied as change moves, so a product's trust posture travels with it (Chapter 5).
- **Products are current because their inputs never went stale.** Operational copies, streams, medallion tables, and grounding products are maintained from the same committed change, on the latency rung the business case warrants (Chapter 4).
- **Execution and persistence are the customer's choice.** Workloads run in GoldenGate, Oracle databases, local or external Spark, or Oracle AI Data Platform Spark, against relational, Iceberg, streaming, and JSON products, in the cloud and region the customer selects (Chapter 6).

Two terms are defined here and reused throughout. The technical category is the *operational-change-native data fabric*: a fabric that begins with committed transactions and business events,

preserves operational context as change moves, and continuously produces governed data for operational, analytical, and AI consumption. The business-facing description is the *operational truth layer*: downstream analytics and AI grounded in authoritative enterprise data, committed change carried with transaction context, controls, provenance, and processing evidence. It doesn't mean the fabric determines real-world truth or validates model outputs; it means that the systems which reason about the business reason **about the business as it is right now**. Launch presentations call the same idea the AI Truth Layer [2].

3.4 What Is New at Launch

GoldenGate Data Fabric extends GoldenGate from real-time data movement into a unified environment for building, governing, operating, and consuming real-time data products.

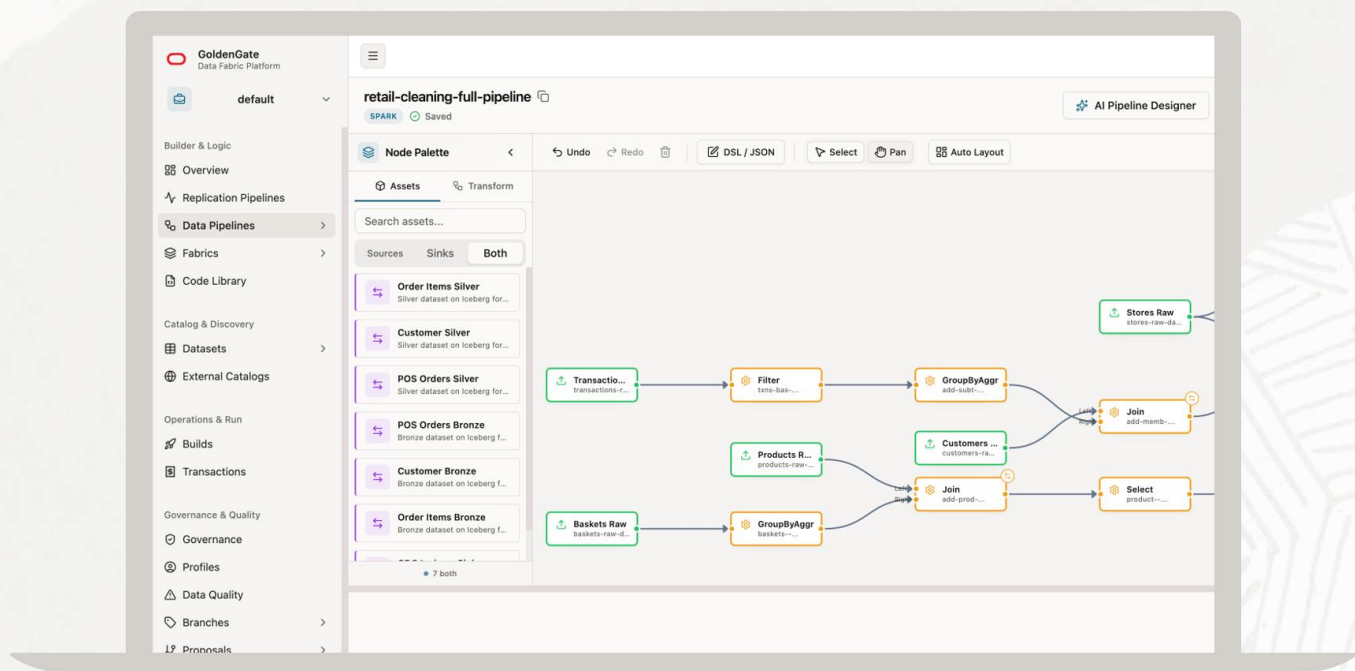


Fig. 3.2. The GoldenGate Data Fabric Workbench: AI-assisted pipeline design over Bronze and Silver Datasets on Iceberg.

Five expansions carry that shift. **One experience** for operational and analytical fabrics, so a replication team's topology knowledge becomes a template an analytics team reuses. **AI-assisted development** at three levels of control (natural-language prompting, low-code editors, a high-code domain-specific API), with every generated design passing human review, testing, validation, and policy checks before deployment. **Governed data products** published as tables, Datasets, streams, APIs, and Gold-layer serving structures, with policy, quality, verification, and lineage evidence attached to the product lifecycle. **Open execution**, with pipelines and products decoupled from any single platform: in a 2026 public list-price model of equivalent data-local Spark pipelines across five managed platforms, the highest estimate ran about 54 percent above the lowest, a modeled spread an enterprise without placement choice cannot act on [18]. **Productized delivery**, with reusable Datasets, patterns, guardrails, deployment controls, telemetry, and estate management turning implementation knowledge into repeatable practice (Chapter 7.4).



The table below shows each core function and the native capability available at launch [2], [3].

Core function	At launch
Data integration	Replication, ZeroETL, ETL, E-LT, CTL; reverse ETL; data preparation; event-driven and batch pipelines
Data products & consumption	Governed Iceberg tables; Datasets; streaming and JSON products; subscription APIs; Gold-layer serving; Delta shares using AI Lakehouse or Databricks
Governance, quality & verification	Policy execution and security controls; validation; verification; compare and repair; profiling; governed publication
AI-assisted development	Natural-language pipeline creation; AI-assisted mapping and transformation; instance-private assistant; built-in or supported external model use
Deployment	Managed Data Fabric service in OCI, Azure, and Google environments; dedicated and sovereign options; management of existing on-premises GoldenGate estates
Runtime engines	GoldenGate, Veridata, local Spark, Oracle AI Database, Oracle AI Data Platform Spark, supported external Spark (Databricks, Amazon EMR, Apache)
Table formats	Apache Iceberg or Oracle native tables (depends on runtime)
Catalogs	Workbench metadata; Oracle AI Data Catalog; Iceberg REST-compatible catalogs
Operations & lifecycle	Design, branch, merge, deploy; pipeline observability and telemetry; fabric visualization; estate management

3.5 What It Is Not

Data management platforms can simplify data and AI within their own environments. GoldenGate Data Fabric connects the operational enterprise across those environments in real time. It is not, by itself, a database, a lakehouse, an enterprise governance platform, an MDM platform, a data marketplace, a BI suite, a model-training platform, or a universal ontology platform, and it does not replace the platforms an enterprise already runs. Its job is to make those investments current and trustworthy: it publishes product metadata, lineage, and quality evidence into Oracle AI Data Catalog and participates with external business catalogs, so enterprise semantics, stewardship, and governance operate on fresher, better-evidenced inputs.

When to Choose GoldenGate Data Fabric

A strong fit when...	Not the primary answer when...
The enterprise must preserve transaction context across operational, analytical, and AI paths	The requirement is only a standalone BI or dashboard platform
Data and applications span multiple databases, clouds, and on-premises environments	The entire workload already lives inside one platform with no external operational integration
Real-time or near-real-time freshness materially affects business value	The workload is periodic and freshness has little marginal value
Customers want governed products without forcing all data into one lakehouse	The primary goal is a universal enterprise ontology or master data platform
Workload placement, sovereignty, and data locality matter	The main objective is a complete model-training or agent-development platform



3.6 The Hero Order Journey: Capture Once

Return to the 9:41 order. GoldenGate captures the complete, ordered transaction once, from the change log, within moments of commit: the order row, the inventory decrement, and the shipment flag together, as the atomic event they are. Now follow the four stages instead of the three pipelines.

1. *Committed change.* The transaction enters the fabric at low latency with its boundary intact.
2. *Real-time events.* The fabric enriches it with current, governed inventory and fulfillment context, applies the product's policy and quality rules, and records provenance and processing evidence.
3. *Governed products.* The standby stays current, an application event fires, a fraud stream enriches, Bronze and Silver Iceberg tables update, and an order-state Dataset refreshes behind a subscription API, each a contracted product with an owner and a freshness SLO.
4. *Trusted action.* The order is routed and the delivery promise confirmed on stock that is actually available, and an AI demand-planning agent is notified with grounding context that reflects the commit. Its reorder writes back as the next trusted event in the fabric.

One capture, many products, one measurable path. GoldenGate Data Fabric keeps the Iceberg tables semantically consistent as they update, so the three changes reach readers as one business event. Where Chapter 1 left the fulfillment agent acting on a partial view of the order, the same agent now acts on a complete view within the product's freshness SLO, and the commit-to-action latency for that journey is a number the customer can baseline this quarter. That is the difference this launch exists to deliver.

The missing layer is a data fabric that spans the operational systems that run the business and the platforms that learn from it. It keeps mission-critical systems in step, turns each committed change into current, governed data products, and keeps pipelines and products portable, so workloads can be placed, and re-placed, wherever the economics work best. GoldenGate Data Fabric is that missing layer.

PART II — The GoldenGate Difference

Chapter 4. One Change, Many Products

Many data estates re-extract the same source data once per consumer: one connector for the warehouse, another for Kafka, a third for the vector store, a fourth for the standby. Every extraction is another source burden, another schedule, another failure mode, and another version of the truth. GoldenGate Data Fabric captures committed changes once, then applies the processing and controls each product needs, so operational teams, analysts, and application developers build on the same business events as their requirements grow.

4.1 Capture once, activate many

One capture can maintain operational, analytical, and AI-facing products from the same committed business change. The 9:41 order enters the fabric with its inventory decrement and shipment flag. Ordered updates keep operational copies synchronized for availability and recovery; application events reach services through Data Streams, the native pub/sub layer based on Secure Web Sockets (WSS) and AsyncAPI. Synchronized state, application streams, and operational APIs are first-class products, with consumers and service expectations of their own.

Operate comes first. Before a committed change feeds a lakehouse or an agent, the same capture keeps a standby in step, carries a migration through without a freeze window, or holds two active sites in agreement. Those replication pipelines are built and governed in the same Workbench as everything downstream, so an enterprise that starts with availability or migration has already built the first half of its fabric.

The same capture feeds Bronze history, conformed Silver tables, and Gold facts and metrics. Enriched event streams support risk analysis; current-state APIs and subscriptions supply applications and agents. Where grounding requires embeddings, a pipeline can call a customer-configured model endpoint under its policy controls and error handling, so vector search reflects recently committed transactions rather than the last batch embedding run (RAG – Retrieval Augmented Generation). Each product keeps its own processing path and freshness expectation. Capturing one complete transaction does not make publication atomic across different kinds of target, such as a standby, a stream, and a lakehouse, so each product is contracted accordingly; within the lakehouse, GoldenGate Data Fabric keeps the Iceberg tables it updates semantically consistent. One capture, many products, each on its own path, and the source is asked only once.

4.2 How a product stays current

Every hands-on practitioner knows the awkward part usually comes at the beginning, well before steady state. Maintaining an order-state product starts with the handoff from initial load to continuous change: GoldenGate coordinates the initial load with a known source log position and



resumes change processing at exactly that point, so there is no gap between snapshot and stream and no double application of changes made during the load, in supported and correctly configured paths, with the procedure determined by source and target [4], [5]. That handoff is the difference between a demo and a production system, and it is solved by the tool rather than left as an exercise for the developer.

From there, Continuous Transformation and Loading (CTL) filters, maps, enriches, and converts changes before publication, so consumers receive data that is current and ready to use rather than raw CDC events awaiting a second job. Streaming runtimes add event-time windows, joins, state, deduplication, aggregation, and routing, with explicit handling for late events and errors, because event time and processing time are different clocks and pretending otherwise is how fraud models miss things. ZeroETL lands Bronze history; CTL maintains conformed Silver and feeds Gold serving structures; batch and micro-batch remain available where they are the better economics. The medallion architecture is not replaced; it is finally fed properly with a continuous stream of trusted data events.

Within GoldenGate Data Fabric the Dataset associates this work with a durable logical product: its schema and interface, freshness expectations, run evidence, and physical-version history. Consumers can identify what they are reading and how it was produced as physical tables evolve. Part III describes the full lifecycle contract and distinguishes launch capabilities from operating commitments and roadmap automation [2], [3].

4.3 Match freshness to the business

Here is a sentence you will rarely find in a launch paper for a new real-time tool: **not every workload should stream**. Real time is a design decision. Choose a rung according to how quickly the value of the data decays and what the consumer can do with it; the top rung, transactional continuity, is about ordering and recovery, not speed.

Table 4.1. The Latency Ladder.

Mode	Appropriate when	Typical pattern
Transactional continuity	The target participates in operational availability, migration, or application state	Ordered CDC and replication; often raw transactions only
Event-driven continuous	Decisions or actions lose value within seconds or minutes	CTL, streaming, event products with full medallion architecture
Micro-batch	Small delays are acceptable and efficiency matters	Incremental transforms and lakehouse maintenance
Batch	The business process is periodic and freshness has limited marginal value	Scheduled ETL or E-LT jobs
Query or access in place	Copying is unnecessary and source performance and governance allow it	Federation or direct open-table access; often requiring a local cache



Source impact, log retention, volume, and processing cost can move a workload down the ladder, and so can the consumer: a team that acts monthly gains little from continuous delivery. A monthly regulatory report does not need a millisecond pipeline; it needs a correct one. The platform's job is to make the right rung cheap, not to make every rung streaming.

4.4 Closing the loop

The loop closes where most lakehouse diagrams stop. Current data products let fulfillment route the order on stock that is actually available and notify a demand-planning agent with the same operational context. An authorized reorder then commits new state in the business system and becomes the next event for the fabric to capture. Application owners keep decision authority, approvals, and action audit; the fabric transports and governs the data on both sides of the loop. Stages of the loop are: *commit* → *capture* → *transform* → *verify* → *publish* → *discover* → *consume* → *decide* → *act* → *observe*



Fig. 4.1. Closing the loop on trusted transactional events to business actions.

This detailed path fits within Chapter 3's four stages: committed changes become real-time events, publication makes governed products available, and consumers use them for trusted actions. The source establishes the commit. GoldenGate handles capture, transformation, supported verification, and publication; catalogs provide discovery; applications and consumers own decisions, actions, and observation of the result.

For the pilot, measure from the source commit until the fulfillment system commits and confirms the

routing decision; that is the endpoint, stated before any number is attached to it. Capture-to-apply timing alone cannot establish it, which is why the six stages from Chapter 1 are recorded separately, fabric processing on one side and consumer and application delay on the other. Baseline the existing journey, agree on a target, measure the result under load. The 9:41 order then has a number, and the number is the customer's.

Chapter 5. Trust That Travels With the Flow

Consumers need evidence that a product is complete enough, current enough, and authorized for its intended use. The trust chain identifies the controls and responsibilities that support that judgment: *Source transaction → capture fidelity → schema and contract → transformation tests → data quality → provenance and lineage → policy → verification → product SLO → accountable consumption*

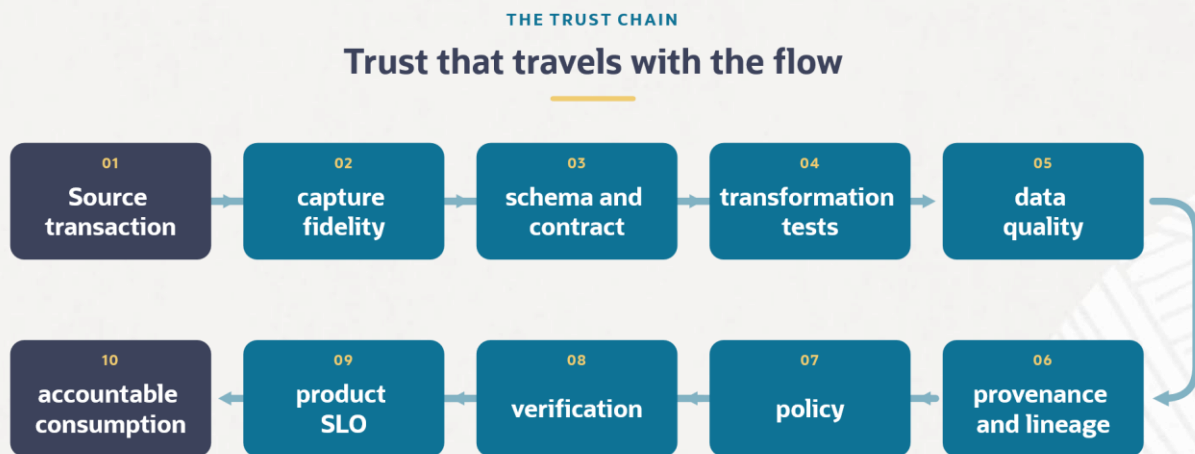


Fig. 5.1. Verifiability and context that travels with the flow of the data.

The System of Record (SoR) establishes the transaction. The fabric supplies capture and processing controls, with Veridata and catalogs contributing their defined services. Product owners define business rules and permitted use; consumers remain accountable for their actions. Each link has an owner; break any link and downstream confidence is borrowed, not earned. Recovery evidence shows how the chain holds through an interruption.

5.1 At capture

Trust starts where authoritative state is established. GoldenGate captures committed changes with transaction boundaries and source ordering intact: Oracle sources supply log-position and commit metadata, other sources use their documented semantics, and downstream processing can interpret what it receives. Resumption without gaps or duplicates and exactly-once delivery require a supported source-target topology and correct configuration.

The JSON schema and API contract define required fields, compatibility rules, and what producers

and consumers may expect of each other under versioning. For supported sources and DDL operations, schema changes travel with the DML so targets stay structurally aligned; coverage varies by source, target, version, operation, and configuration. Within that coverage, compatible changes propagate, while breaking changes are typically caught at design time, tested, and promoted deliberately, with rollback where available and consumer notification under the contract policy, not discovered by a consumer at 2 a.m. Schema change stops being the thing that silently breaks the estate on a Thursday.

5.2 In motion

Validation and policy controls decide what can enter a published product and who may use it. Native launch controls include profiling, completeness, type, range, and referential checks, with deduplication and business rules for the product's requirements; failed records go to quarantine or error handling, with human review where required. Identity, encryption in transit and at rest, masking, filtering, promotion approvals, and audit apply as the data moves, before it reaches the destination, and data access follows the customer's identity and policy configuration. Sensitive-data rules ride the pipeline: an analytical consumer receives masked fields while an authorized operational consumer keeps the fidelity its task requires [2], [3].

Pipeline-level lineage and operational evidence are native at launch. Oracle AI Data Catalog supplies integrated discovery and context; enterprise remediation and business lineage remain with the selected governance platform; end-to-end lineage across external systems and automated lifecycle controls are roadmap direction (Chapter 3.4). What the fabric records, the Dataset carries: source provenance, transformation and pipeline-run history, deployment history, and physical versions travel with the product, so a product's policy posture and its evidence are properties of the product everywhere it is consumed.

5.3 Verification and recovery

Next comes the link most pipelines do not have. Pipeline monitoring answers proxy questions: did the job run, were there errors, how many rows moved. For supported comparisons of a source-target copy, Veridata provides auditable proof: it compares source and target data, accounts for concurrent replication, identifies out-of-sync records, and supports reporting and repair where applicable [6], [7]. The Veridata comparison evidence establishes reconciliation for an operational copy; transformed products need their transformation tests and quality checks as well, which may be encoded into validation pipelines.

Verification is the difference between hoping data arrived correctly and being able to prove it did. Quite useful for compliance and audit!

Suppose the target connection fails while the order-state product is being updated. In a supported configuration with the required trail data retained, the endpoint owner restores availability and the runtime resumes from the recorded checkpoint, with apply behavior following the target's exactly-

once or idempotency requirements. The product owner checks checkpoint progress, run evidence, and restored freshness; Veridata comparison can add reconciliation evidence for the source-target copy. Nothing about that recovery depends on anyone's memory of what happened; it is in the evidence.

The verification chain ends with a promise consumers can hold: freshness, completeness, availability, schema stability, quality, permitted use, and an escalation path, per product, measured and visible, with product owners recording the clauses not yet enforced automatically. Each of those clauses is a number somebody can check. Where recovery needs more than a checkpoint, such as a replay, a rebuild, or a backfill, the procedures are described with Day-2 operations in Chapter 7.4.

5.4 The operational truth layer – what it means

The operational truth layer grounds decisions in authoritative enterprise changes from the System of Record (SoR) with evidence consumers can examine. GoldenGate preserves transaction context, applies controls as data moves, records processing history, and publishes governed products. To be clear, it does not independently establish real-world truth or validate model outputs. That is the whole point of an operational truth layer: it is strong where it is operationally provable.

AI-ready data meets measurable acceptance criteria for freshness, completeness, consistency, schema compatibility, semantics, provenance, policy compliance, product SLO attainment, and authorized access. Verifiable evidence can take several forms:

Trust evidence by type

- **Measured:** capture-to-apply latency, checkpoint recovery behavior, quality scores, and SLO attainment.
- **Verified:** Veridata comparison results and identified out-of-sync records for supported comparisons [6], [7].
- **Inspectable:** delivered transaction metadata, pipeline-run history, and quarantine queues.
- **Documented:** schema contracts, policy rules, promotion approvals, and audit trails.
- **Conditional:** catalog discovery and context depend on the catalog integration; end-to-end fabric lineage and policy-aware lifecycle automation are roadmap direction.

Chapter 6. Choice Without Chaos: Portable, Open, and Sovereign Data

“Open” has become the most abused word in data infrastructure, so this chapter turns it into a specific concept tied to concrete value. GoldenGate Data Fabric separates logical product identity and governed design from the physical implementations beneath them, so customers can change placement as requirements change and keep what they built.

6.1 Governed products across platform choices

The Dataset preserves product identity across physical implementations, retaining schema and endpoints, processing evidence, and version history. Data products run on the platforms you choose, and four choices stay decoupled from any one platform's tooling:

1. **Where data persists:** databases, lakehouses, and object stores in customer-selected clouds and regions.
2. **Where processing executes:** GoldenGate, database pushdown, or supported Spark engines.
3. **Which catalog manages open tables** and supplies discovery or business context.
4. **Where the service control plane runs:** with data planes placed locally beside the data.

The portability contract states what must hold when those choices change: equivalent operators and functions, SQL and expression behavior, time and state handling, null and ordering semantics, checkpoints and recovery, schema evolution, catalog integration, identity and security, observability, and operational SLOs. There is no magic here. Oracle estimates that for typical pipelines roughly 70 to 80 percent of operations are portable as designed, where operators and semantics are equivalent, and the rest may need a light redesign. Where all clauses hold, full portability is real. Where some hold, it means redesign-lite. Where few hold, the right answer is to say so. Openness reduces friction; it does not repeal engineering, and a stable product identity does not guarantee unchanged execution on every engine.

Business requirements to keep the data product's analytical processing in its home region can change its engine and target while preserving its logical identity and intended contract. In the modern era of multicloud solutions, the home region may actually be a different cloud provider (e.g., as with Oracle Multicloud, including GoldenGate Data Fabric). Before promoting or sharing the data product, the fabric team revalidates operators, access, recovery, and freshness, and records the physical version.

6.2 Where transformation executes

Where transformations run changes performance and cost materially, so the GoldenGate Data Fabric Workbench treats engine selection as a first-class design decision, weighed against latency, locality, volume, statefulness, cost, and team skills. Options include GoldenGate, SQL pushdown into a source or target database, local Spark, Oracle AI Data Platform Spark, and external Spark such as Databricks, Amazon EMR, or Apache; design reuse across them depends on operator support.

Storage and catalog choices need the same care, and the medallion pattern does not settle them. Iceberg provides schema and partition evolution, snapshots, time travel, and multi-engine access. By itself it does not preserve a source transaction atomically across separate tables or ensure identical engine behavior; GoldenGate Data Fabric ensures semantic consistency during the Iceberg table updates it performs. Database-native medallion products remain an option where concurrency, updates, transactional requirements, or security favor that path. Three patterns cover most estates, and the figure below states what each is chosen for and what each obliges you to plan for.

Iceberg throughout Bronze, Silver, and Gold is the right call when large histories, analytical scans, and a choice of engines dominate; plan for updates and deletes as file rewrites or read merges, budget for compaction, and benchmark concurrency before promising it. Serving thousands of simultaneous short reads and writes is a different job, and a mature operational database was built for it: connection management, indexing, concurrency control, and transaction processing tuned for sustained application traffic. Spark over Iceberg delivers excellent throughput on large analytical work; it was not designed for demanding operational concurrency or predictable response times.

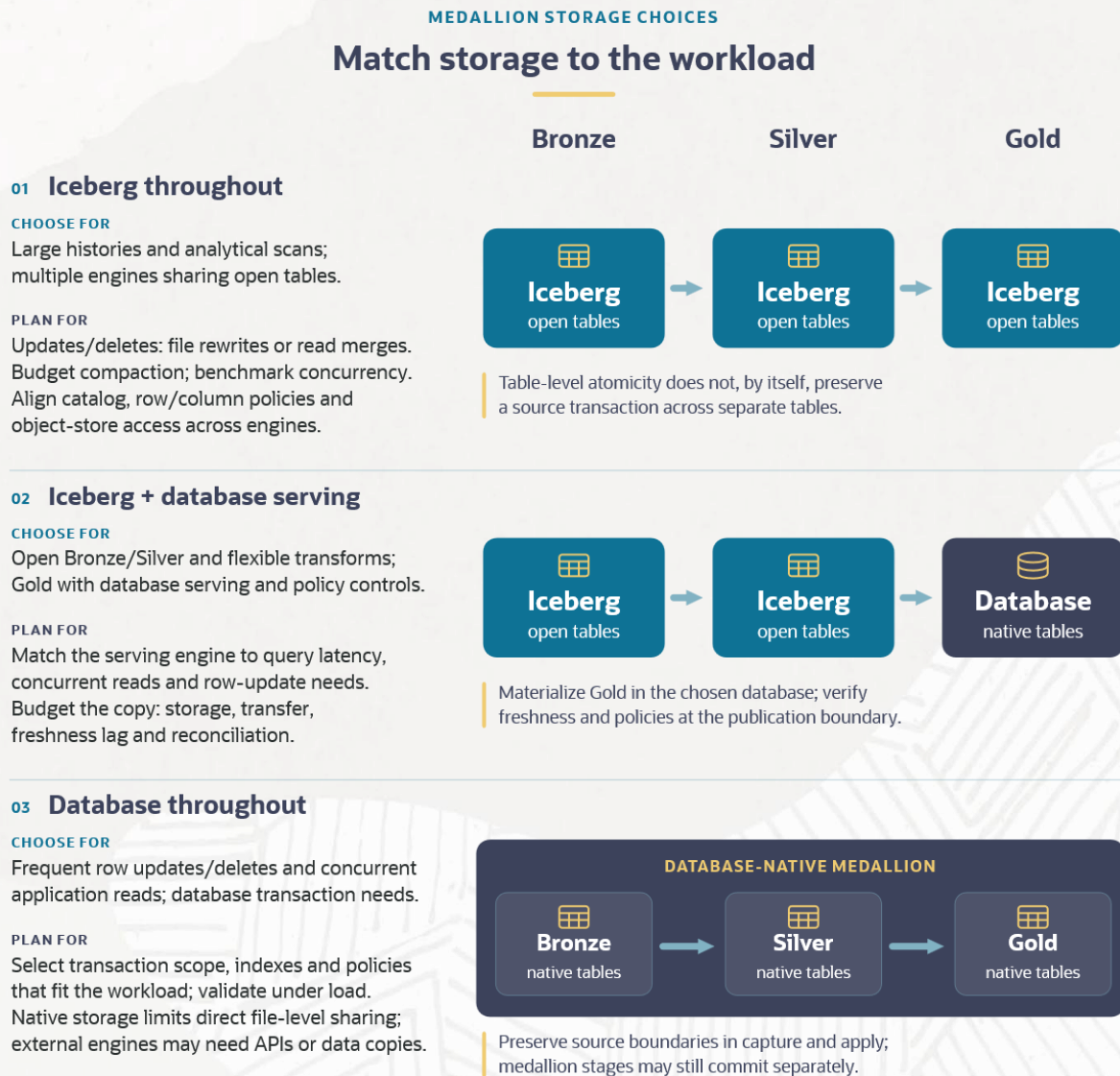


Fig. 6.1. Medallion storage choices: match storage to the workload.

Throughput and concurrency are different virtues, and no storage format supplies both for free. The practical hybrid keeps Iceberg for Bronze and Silver and publishes Gold into a database, preserving engine choice upstream while giving applications and agents a serving layer designed for their demands. Choose your data copies wisely: each one adds storage, transfer, publication lag, and reconciliation to budget for, and freshness and policy need verifying at each publication boundary.



Security is the other reason to consider database serving. Mature enterprise databases layer authentication, roles and privileges, fine-grained access control, auditing, encryption, and separation of administrative duties, and their authorization models often map directly onto the ones already governing the systems of record, which makes policies easier to align and administer, particularly where source and target share a platform. Iceberg provides table semantics over data files, commonly Parquet in object storage; comparable protection is achievable, but it has to be enforced in concert across the catalog, the processing engines, the identities, and the storage permissions. Iceberg can be secured to the same standard. It cannot be secured in one place.

Database-native products across all three layers are therefore a strong option when frequent updates, strict transactional requirements, or alignment with established security controls matter end to end. Preserve source transaction boundaries explicitly through capture and apply, remember that medallion stages may still commit separately, and accept the greater dependence on database-native storage and interfaces, since external engines may need APIs or copies rather than shared files. The question is never Iceberg or database. It is which workload each layer must serve, and which trade-offs you are willing to make on speed, consistency, security, and serving.

Workbench metadata identifies fabric assets. Oracle AI Data Catalog supplies integrated discovery and context; Iceberg REST-compatible catalogs manage open tables. Semantics are linked to Datasets and can materialize as Apache Ossie-compatible facts and measures (when leveraging Oracle AI Lakehouse or Databricks as serving layers).

6.3 Locality, economics, and sovereignty

Workload execution near to the data is what makes platform choice pay dividends. Moving a data-heavy workload across clouds can cost more in egress than the compute it saves, so portability is worth having when execution stays local, when I/O is small relative to compute, when transfer charges are contractually removed (e.g., as with Oracle Multicloud), or when the source systems themselves come from many locations outside of the desired home region. Chapter 10 explores the economics of this more closely, including how actual completed-job cost depends on efficiency, utilization, discounts, and locality [18].

Sovereign placement depends on storage and execution locations, control-plane availability, support for named political boundaries, and network paths, and the fabric lets you set them independently: dedicated and sovereign deployments at launch, private connectivity to existing on-premises estates, and the control plane in the environment your regulator expects. Sovereignty is a configuration, not an exception process.

Portable designs and local execution let customers reassess data platforms as product requirements and economics change, without reassessing the data products themselves. Standardize the fabric, not the platform.

PART III — Putting GoldenGate Data Fabric to Work

Part II stated the principles. Part III puts them to work: one managed product, a shared place to build, publish, and operate data products across the systems an enterprise already owns, and a way to fit it into the estate without rebuilding the estate. The buying case is pragmatic. Teams can put current operational data to work sooner, make data fitness visible to the people who consume it, and reuse proven real-time designs as demand grows.

Chapter 7. From Design to a Governed Product

Enterprise IT leaders are tired of architectures that need six products and a services bench to become usable. GoldenGate Data Fabric is a cohesive managed product with explicit boundaries that augment existing data platforms. This chapter follows one product from design to production and states, at each step, what the product does, what Oracle operates, and what the customer keeps.

7.1 One Experience, Several Ways to Build

The GoldenGate Data Fabric Workbench is the design surface. It is a tool for GoldenGate operations, data engineers and data product managers. It provides an estate view for real-time replication, visual pipeline design, AI-assisted prompting, asset and template libraries, testing, branching and merging, deployment, observability, and discovery of published products.

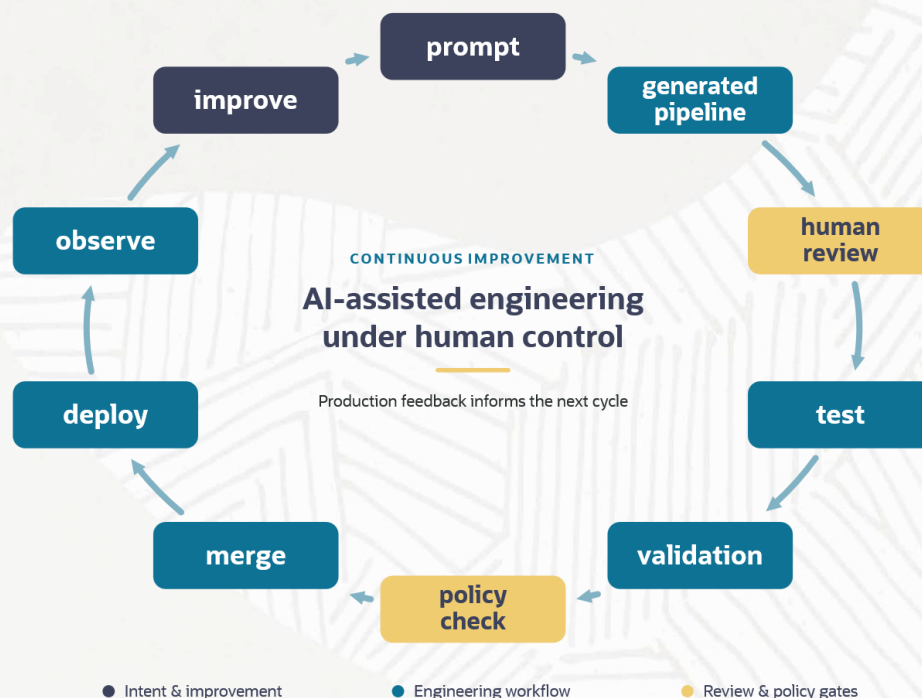


Fig. 7.1. The AI-assisted design loop.



Three levels of control open onto it: natural-language prompting for the analyst who wants to describe the pipeline, a low-code editor for the engineer who wants to see it, and a pro-code DSL, with API and MCP services where supported, for the team that reviews everything in a pull request. Every visual design has a declarative code representation, so a prompt-first analyst and a code-review-first engineer work on the same asset, and a design that succeeded becomes the template for the next one rather than a memory [2], [3].

Take an engineer preparing the order-state product. The assistant proposes source-to-target mappings and transformation logic; the engineer reviews them, tests representative changes, and checks sensitive-data handling before promotion. The assistant is a private process attached to each fabric instance with a user-configurable model. It shortens the path to a reviewed design; it does not shorten the review. Business meaning, access decisions, and release approval stay with people, and every stage of the loop leaves actionable evidence.

One Workbench. Runtimes across locations.

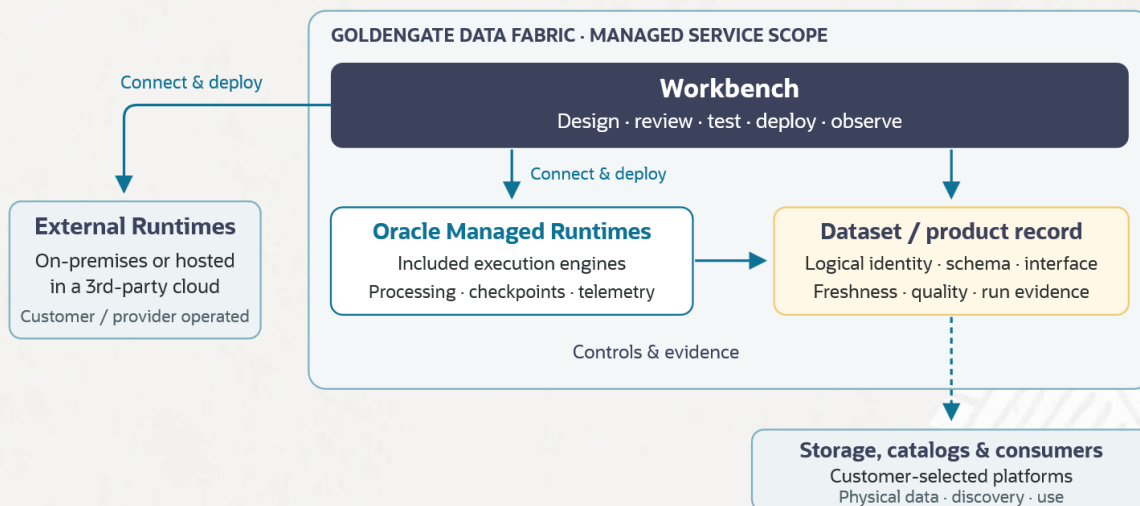


Fig. 7.2. GoldenGate Data Fabric connects design, supported execution, and product evidence while preserving customer choices for storage and consumption.

The Runtime executes beside the data. It plans and schedules execution, processes events, manages checkpoints, scales, handles failure, captures operational metadata, and publishes products through the GoldenGate, ETL and E-LT, local Spark, Data Streams, and ZeroETL engines. Identity, encryption, policy, validation, verification, and telemetry controls span both surfaces. At launch the Workbench runs from OCI GoldenGate and can connect to existing on-premises GoldenGate estates or OCI-GoldenGate managed deployments. That flexibility is what lets a customer introduce the fabric around the estate it already has and place processing where the workload needs it, without having to move every source and consumer onto one platform.

7.2 Many Hands, One Asset

A data product is rarely just one person's work. Three core personas use the Workbench at launch. The GoldenGate architect knows the mission-critical source topology and builds the replication and CDC pipelines. The data engineer builds the transformation and medallion pipelines that turn captured change into Silver and Gold. The data product manager defines the Datasets, the quality rules, and the catalog entries that consumers will hold them to. Testers and administrators work around them, as they do on any data platform, and a small team can fold every role into one or two people without changing the workflow, because the workflow belongs to the asset rather than to the org chart.

Collaboration runs on branches, the way software teams already work. Datasets, quality rules, and the pipeline itself are defined and tested in a feature branch on a test runtime. A merge proposal goes to an administrator, and only after the merge is the pipeline built on the production runtime against production connections. Software development lifecycle management (SDLC) is built into the tool so that nobody reaches production by accident.

A merge should be a decision, not a side effect.

Repeatable patterns are built into the Workbench as well. GoldenGate Recipes package the common replication designs, including one-way replication, active-active synchronization, and migration, as reusable templates: a team picks the pattern, supplies endpoints and mappings, and materializes it as a configured GoldenGate pipeline, so established practice arrives with every deployment rather than being rebuilt from memory [20]. ZeroETL takes on the part of setup most likely to go wrong by hand, target instantiation. It coordinates the initial copy with the handoff to continuous change replication, so the target starts from a consistent baseline and stays current from that point on [21]. With an expanded range of supported sources and targets behind both, the path from choosing a business pattern to running a live data flow gets shorter, and the setup that used to be repeated per project stops being repeated.

The Workbench's Replication Fabric brings the whole distributed GoldenGate estate into one topology view. Teams register deployments wherever they run, on-premises and in sovereign, federal, or commercial cloud regions, connect them through authorized endpoints, and see and govern the Replication Pipelines configured between them. Deployments can connect directly or relay changes onward through intermediate deployments along configured paths, the cascading topologies GoldenGate has long supported [22]. Mesh Wi-Fi is the right visual picture: each connected node extends the reach of the whole, and here the reach is operational data movement across physical locations. The Replication Fabric editor makes those relationships visible, so a team can trace how a change reaches its consumers and see where a new pipeline belongs. Distributed replication is governed as one estate, while deployment choices still follow locality, connectivity, and regional requirements. One team, two pipeline types, one promotion path.

Underneath the workflow, GoldenGate Data Fabric runs as a secure multitenant service with a dedicated data plane per customer, isolated to the customer's tenancy and compartments, for service routing, networking, and private endpoint connections. At launch, Developer and Administrator roles separate the people who build from the person who promotes. Roadmap direction adds a scoped



tester role that can exercise pipelines but not change them, production-flagged connections with their own permissions, and paired test and production connections for replication pipelines, so that promotion no longer means editing the pipeline.

7.3 Publish Products Others Can Use

A Gold table with a nice name is not a product. A pipeline is how data moves; a table is what landed; a data product is a commitment: this data, with this meaning, at this quality, at this freshness, for these consumers, owned by this person, until formally retired. The fabric expresses that commitment in a product contract and enforces its measurable checkpoints in the flow.

A data product is a commitment.

A data product contract lives on the Dataset introduced in Chapter 3: a durable logical identity that connects the data's meaning and access interface to its schema, source, quality rules, measured freshness, run evidence, and physical-version history. Physical tables can change engine, region, or format while consumers keep one point of reference for what they are reading and how it was produced [2], [3].

The 9:41 order, owned by the Order Platform team in the Commerce domain, feeds an order-state API with a freshness SLO in seconds, an ordered fraud stream retained for seven days, a Silver conformed orders table, and an agent-grounding product with PII masked for analytical consumers and full fidelity for authorized operational ones. Quality rules catch a missing order identifier or an invalid status before either reaches a consumer. Catalog entries can record who may use it and how change will be communicated, so a consumer can judge whether it is fit for a service workflow, an analytical model, or an AI grounding use case. One source of truth, four contracts, one accountable team, zero duplicate extraction projects.

The distinction between a recorded commitment and an enforced control matters. Schema, source, interface, core metadata, quality rules, freshness measurement, run evidence, and version history are critical capabilities. Enforcement can happen in the fabric, through Data Streams, or in the serving platform, where customers set lakehouse policies on the Gold layer or on a Delta Share. The contract records the commitment in either case, and the control does not have to sit inside GoldenGate Data Fabric to be enforced. Ownership and discovery use Workbench metadata and Oracle AI Data Catalog integration.

This Workbench operating model borrows the best of data mesh without its sprawl: a central platform team owns connectivity, controls, and patterns; domain teams own meaning, pipelines, fitness, consumers, and lifecycle. The GoldenGate Data Fabric publishes into catalogs rather than running yet another marketplace of its own.

7.4 Keep Products Useful in Production

Online practitioner forums are unambiguous: data operations are a purchase criterion in their own right. A consumer needs to know whether a product is current and dependable today, and an

operator needs to be able to distinguish a late product from a failed flow from a data-quality problem. The fabric gives both teams the same evidence.

Operating a product means lag and freshness monitored per pipeline and per product SLO. Schema change is triaged against consumer contracts, with guided promotion or rollback. Replay and backfill, whether a transaction replay, a rebuild from Bronze history, a historical backfill for a new consumer, or reprocessing after a logic change, each carry their own retention, checkpoint, and idempotency procedure, and a recovered job is not finished until the product's service level is restored. Dependency failures surface with cause rather than symptom; capacity and cost anomalies are flagged against expected run profiles; and source log pressure and retention are watched continuously, because they are the quiet killer of CDC data replication estates.

Operations responsibility follows the service boundary in Fig. 7.2. Inside the managed service scope, Oracle operates the Workbench, the Oracle-managed runtimes (lifecycle, execution, checkpointing, and telemetry), and the included GoldenGate, Veridata, and local Spark engines, and it handles fabric service incidents per SLA. External runtimes sit outside that scope: customer-operated GoldenGate deployments and external Spark clusters, on-premises or in a third-party cloud, are run by the customer or its provider even when the Workbench connects and deploys to them. Customers and third parties also own data content, product ownership and business logic, networks and endpoints, and the lakehouses, warehouses, and catalogs the fabric publishes into. Shared items, including upgrade windows, private routing, design review, promotion approvals, and cross-boundary triage, are a part of the shared operating model.

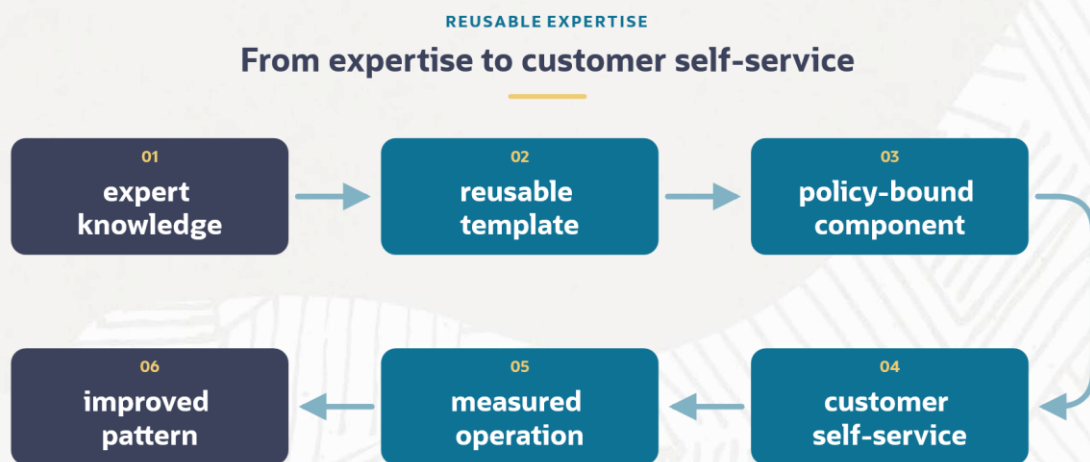


Fig. 7.3. From expertise to customer self-service.

That shared process flow is the answer to the services-dependency trap practitioners rightly fear. Implementation knowledge from GoldenGate veterans, data engineers, and the platform team becomes templates and policy-bound components, so each new use case needs less outside help than the last. Domain teams design, test, and deploy within those templates; platform-team approval gates connectivity, sensitive-data policy, and production promotion; deep replication topology and recovery design remain expert tasks. That is self-service with guardrails.

Chapter 8. Fit the Enterprise You Already Have

Enterprises have already paid for their operational systems, their data platforms, and the people who run them. GoldenGate Data Fabric earns its place by extending the value of those investments and connecting them to new outcomes. This chapter covers where it fits, which catalog it publishes into, how it coexists with a strategic platform, and where to start.

8.1 Three Solution Patterns, One Connective Layer

Three starting points cover most customers. These are solution patterns that can be combined, and GoldenGate is the connective layer under all of them [2], [14], [17].

Customer situation	GoldenGate contribution	Customer value
Existing operational systems and GoldenGate deployments	Extend supported capture and delivery flows into governed data products.	Established connectivity and operating expertise start serving new consumers.
Oracle data, analytics, and AI services	Supply current operational data to Oracle AI Data Platform and Autonomous AI Lakehouse, where supported.	Operational change reaches analytical and AI use cases inside the Oracle estate.
Heterogeneous databases, lakehouses, clouds, and catalogs	Design and deliver governed flows across certified combinations.	Platform choice kept; repeated integration work retired across the estate.

Most enterprises will run more than one of these at once; the fabric is what keeps them coherent.

The Oracle ecosystem platforms keep distinct roles. Oracle AI Data Platform supplies analytics and AI development; Autonomous AI Lakehouse supplies the Autonomous-based data warehousing environment; Oracle AI Data Catalog supplies discovery and context; customer and third-party platforms supply whatever they were chosen for. GoldenGate Data Fabric supplies the operational integration, continuous transformation, validation, and delivery that keep all of them current. Each layer gets better data because the fabric exists, and the fabric stays honest because it does not pretend to be the layers above it.

Oracle AI Database depth strengthens the case for Oracle customers: the private, Oracle-native capture is Oracle's strongest option within that real-time scope. Heterogeneous connectivity extends the same fabric across a wide array of supported non-Oracle sources and targets, including the most popular data platforms such as Databricks, Snowflake, Google BigQuery, Microsoft Fabric and various Amazon data services. That depth is a strength of the fabric and puts no limit on its scope [4], [5], [14], [15], [17].



8.2 Bring Your Own Catalog

The catalog is where an estate keeps its memory: its authorization model, its audit controls, its discovery, and the external engines that read its tables. Replacing it creates friction and undercuts open execution, so the fabric does not ask to. GoldenGate Data Fabric Workbench includes its own internal metadata catalog for important internal objects. It also runs Oracle AI Data Catalog for table metadata and Apache Iceberg compatibility, and it can integrate with a range of third-party catalogs as well. The value of a data fabric is in the portability of governed, logical Datasets and their lifecycle, not in the ownership of every physical table catalog, so customers can bring and integrate their own catalogs if they choose.

For customers who want an Oracle-managed catalog, Oracle AI Data Catalog can be embedded in the fabric service, deployed separately, or hosted within Oracle Autonomous AI Database as an Iceberg REST catalog [17]. Catalog choice stays independent of where the fabric runs. The overall GoldenGate platform can integrate with any of: AI Data Catalog, Hadoop Catalog, AWS Glue Data Catalog, Project Nessie, Snowflake Polaris Catalog / Open Catalog, and standard REST Catalogs (including S3 Tables and IBM watsonx.data).

Customer profile	Likely catalog pattern
Established OCI and Autonomous AI Database customer	Autonomous-hosted Oracle AI Data Catalog; the fabric publishes governed Iceberg products into the existing Oracle data platform.
OCI customer needing independent catalog ownership or shared production services	Separately deployed Oracle AI Data Catalog.
Greenfield customer	Embedded catalog for rapid onboarding; promote to a separate or Autonomous-hosted catalog when shared production requirements emerge.
Amazon, Azure or Google associated fabric customer wanting an Oracle-managed Iceberg catalog	Separately deployed Oracle AI Data Catalog with integrations to Glue or other REST compatible catalogs.
Customer standardized on Glue and Lake Formation, Unity Catalog, Polaris, or another enterprise or open-source Iceberg REST catalog	The customer's catalog stays the production default; the fabric publishes into it via tight integration.

The embedded catalog is most useful in development and testing: isolated, repeatable environments that leave enterprise catalog structures alone. In production it fits when a product is managed and consumed entirely through the fabric. When several teams, engines, or domains need one shared production catalog, a separate or Autonomous-hosted Oracle AI Data Catalog is the better answer.

Bring the catalog you already trust.



8.3 Start with One Product and Expand

Start with a business outcome that needs better operational data and has a named data consumer: a more current service view, an earlier exception signal, more dependable AI grounding. Define what that consumer needs, agree on the product's meaning and operating expectations, and use an existing source and delivery path where one exists. A first investment that is specific, easy to evaluate and easy to sponsor.

Expansion of your data fabric follows demonstrated value. A second consumer reuses the product; a different policy, interface, or freshness requirement calls for a related product built from the same capture and design patterns; each extension is justified by another business outcome. Adoption goes product by product.

8.4 Coexist, Extend, or Consolidate

The six archetypes defined in Chapter 2 narrow to four core coexistence patterns:

1. feeding an external data platform-centered suite's Bronze layer while operational copies stay outside it; GoldenGate is the connective tissue between Ops and Analytics
2. supplying fresh, evidence-bearing transaction flows to a data governance and AI context platform; GoldenGate provides the real-time data streams with trusted data events
3. pairing federation for exploratory data discovery with the data fabric for ordered, durable propagation of data
4. delivering authoritative operational events into an ontology and agent layer over an open data platform that uses open data formats; GoldenGate is the foundation for the operational truth layer that grounds modern AI/ML in authoritative, verified data

When to complement, extend, or consolidate depends on operational use cases, source heterogeneity, latency, sovereignty, reliability requirements, and existing investments. It is a decision to make case by case.

Nothing in this chapter asks you to abandon a strategic platform. Everything a fabric adds makes that platform more current, more future-proof and also easier to evolve.

PART IV — Journeys, Economics, and Proof

Chapter 9. A Lighthouse Journey

Architecture persuades architects. Journeys persuade everyone else. And the hardest test of any data fabric is not the first pipeline; it is the fifth platform. What happens when the warehouse gets too expensive, the data lake gets too hard to secure, the data center closes, and then, years later, some of it comes back? One anonymized journey, drawn from a Fortune 100 consumer electronics company, answers that question across nearly two decades. Platforms came and went on both the operational and the analytical side of the house, each for a good reason at the time. GoldenGate stayed, and every new destination built on the change streams already serving the business.

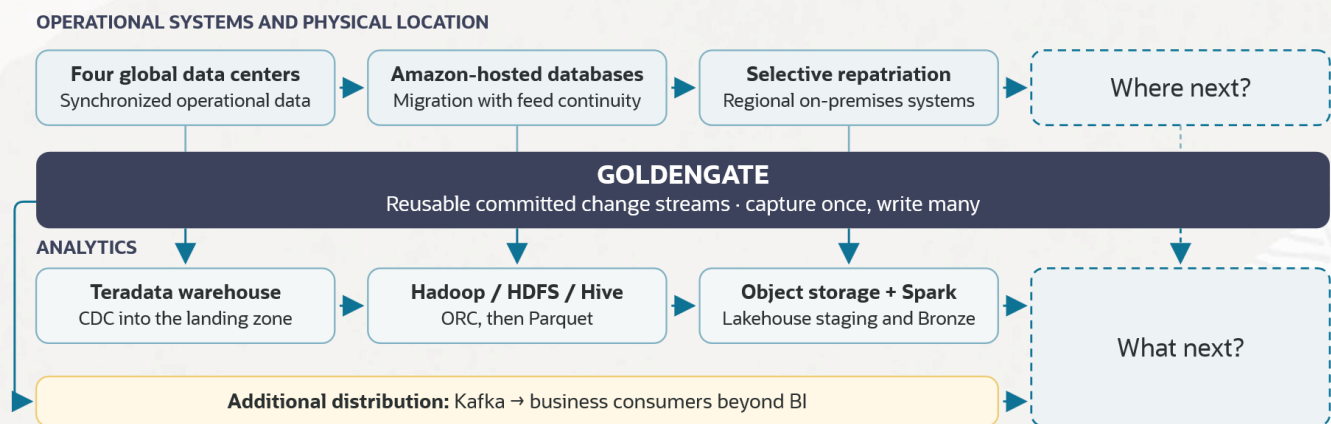


Fig. 9.1. Platforms and physical locations evolve; reusable change streams provide continuity. The operational and analytical transitions overlapped.

9.1 One Customer, Several Generations of Platforms

It began in the early 2000s with an operational problem, not an analytical one. The company ran a global consumer business with users active on every major continent, and it needed to synchronize and act on that behavior across four major data centers around the world. GoldenGate kept the operational databases in step: user behavior, transactions, and account state replicated across continents while the business ran on top of them. Analytics showed up years later and found the streams already running.

When the company built its first global warehouse on Teradata, it wanted real-time ingestion and had no appetite for another batch ETL path. So it tapped the GoldenGate CDC streams it already had and fed the Teradata landing zone directly. That worked for years, right up until Teradata's storage costs stopped working. The analytics repository was rebuilt on Hadoop, HDFS, and Hive; the change streams stayed where they were and began landing in ORC, later Parquet. The BI teams rewrote their jobs for Hadoop. Nobody rewrote the extraction.

Hadoop fixed the storage bill and created a different problem. Other lines of business wanted the

user-behavior data and the aggregates built on it, and the IT team was not willing to hand them Hadoop's access-control posture or its thin options for remote consumption. Kafka became the distribution and consumption layer, and GoldenGate fed it from the same operational sources it was already capturing. Consumers multiplied. The capture path did not.

While the analytics estate churned, a larger shift was underway beneath it. Over several years, most of the operational databases moved off owned hardware and into cloud-hosted databases on Amazon. GoldenGate ran the zero-downtime migrations, and the same GoldenGate kept the change feeds into the warehouse and Kafka flowing throughout. Operational continuity and analytical continuity, from one tool, across a multi-year capex-to-cloud transition.

Overlapping that move, the analytics teams left Hadoop for what we now call a lakehouse: data on object storage, processing decoupled from it and running on Spark. The GoldenGate streams that once landed in Hive now land in the staging and Bronze tiers of a medallion lakehouse. That remains the foundation of this customer's analytics at scale today.

The most recent turn runs the other way. For some of the largest, most mission-critical OLTP systems, the layers of virtualization and software-defined networking beneath a cloud-hosted database added more I/O overhead than the performance and latency targets could absorb, and optimized bare metal in regional on-premises data centers simply did the job better. GoldenGate carried those systems back out of the cloud with the same care it had carried them in. Operational placement and analytical platform choice moved on separate schedules, in both directions, and the business ran through all of it.

Five platform generations. Two migrations in opposite directions. One capture path, and not a single system-of-record extraction rewritten.

9.2 Extend the Continuity Beyond Bronze

Nobody can say what this customer's next platform will be. Everybody knows that there will be one. That is the real lesson of two decades: not that the company chose the right platform every time, but that choosing again never cost it the foundation. The story reduces to two facts. First, GoldenGate as a fabric let the company re-optimize its operational systems and its analytical platforms several times each without disrupting the core business. Second, in all that time it never had to rewrite an extraction job from a system of record. Capture once, write many populated whatever the staging layer happened to be that year, and the staging layer changed a lot.

Standardize the fabric, not the platform.

That history shows what durable capture and continuity through platform change are worth; it predates the Workbench and the Dataset. GoldenGate Data Fabric takes that principle further than staging and Bronze. The Workbench, reusable pipeline designs, and the durable Dataset carry the same continuity into transformation and the product lifecycle: the product's meaning, controls, and operating evidence stay put while supported execution and storage choices change beneath them [2], [3]. A platform transition still takes engineering and validation; nobody should pretend otherwise. What changes is how much of the investment travels with the customer, the answer is now most of it.



Return to the 9:41 order one last time. The next consumer of that order might be a service application, a supply-chain exception process, a fraud investigator, or an event-driven AI agent. Each reads a governed product through the interface it needs, and none has to wait for the data to make a stop in an analytical platform first. One committed change, several business outcomes, and every new consumer inherits the work already done.

Chapter 10. Economics and Measurable Outcomes

10.1 The Economic Model

The Chapter 9 journey is the economic case told as history. Teradata's storage bill went away without a re-extraction project. Hadoop's access problem was solved by adding a consumer, not a capture. Two migrations in opposite directions ran without an outage, and the analytics feeds never noticed. Each of those is a line item somebody else's estate paid for in full: a rewrite, a freeze window, a parallel extraction path that lingered for years. GoldenGate Data Fabric applies the same arithmetic to more of the pipeline, so the business case has to count two things: what today's operation costs, and what the next change will cost.

Three benefits belong in that calculation. Reuse: fewer duplicate integrations and less engineering for the next product or consumer. Operating efficiency: execution placed where it belongs, recovery that is dependable rather than heroic, and less time reconciling copies that disagree. Business responsiveness: a current service view, an earlier exception signal, a better-informed agent. Which of the three matters most depends on the journey, and the resulting number should be legible to the business owner as well as the platform team.

Then count the whole cost of delivering that outcome, beyond the compute line: fabric and platform charges, compute, storage, network transfer, catalog services, engineering and support effort, and the temporary coexistence every transition requires. Count retired work and released capacity where they can be shown, and resist booking every reused pipeline as an immediate cash saving. The strongest buying case pairs an accurate operating baseline with a credible next use case.

10.2 Put Workloads Where the Economics Work

Platform choice is worth money only when execution can follow the workload. The Chapter 9 customer learned that the expensive way, on the repatriation leg: for its heaviest OLTP systems, the right placement was the one with the least between the database and the disk. In the 2026 public list-price model this paper cites, the highest estimated monthly cost for three data-local Spark workloads ran roughly 54 percent above the lowest across five platforms, with runtime held equal and the Oracle catalog allocation included. That is a placement opportunity, not a benchmark, and it should be read as one [18].

Data locality comes first, and it is not close. In the modeled 500-GB workload, cross-cloud transfer cost roughly 37 to 50 times the compute cost of the run itself. Compare completed-job cost beside



the data, with actual utilization, runtime efficiency, discounts, and existing capacity commitments in the math, because any of those can reorder the ranking. Reference [18] holds the model and its assumptions. The decision is which supported placement gives this customer the best cost at the required service level [18].

Keep the pipeline reusable. Put execution where the workload belongs.

10.3 Measure the Next Outcome and the Next Change

Keep the scorecard small: one new number and six that most teams already track. The new one is commit-to-action latency, the elapsed time from a system-of-record commit to a trusted business response, and it is the headline number for any operational or AI product. The six give it context. Freshness SLO attainment and reconciliation gap show whether the product can be trusted. Time-to-pipeline and pipelines per engineer show whether delivery is getting faster. Consumers per product shows reuse, and cost per workload shows what it all costs. For a modernization journey, add cutover interruption and the effort to reconnect consumers.

The Chapter 9 customer suggests the one test that matters most: what has to be rebuilt when a destination changes or a new consumer arrives? Stand up a second delivery path, or hand an existing product to a new consumer, and portability stops being a slide. Give every chosen metric a baseline, an agreed target, a measured result, and a named owner. That is the line between a good demo and an investment case.

10.4 The 90-Day Change-to-Action Assessment

The Assessment turns the buying case into a decision on the customer's own numbers, and it runs as a trial of GoldenGate Data Fabric: one valuable journey, one accountable business sponsor, a short list of success measures, and scope agreed around available access, supported endpoints, and the customer's delivery calendar. Ninety days later there is a working product and evidence of reuse.

Phase	Business focus	Evidence for the next decision
Baseline Days 1–15	Choose the journey, consumer, and product commitment.	Current latency, cost, and effort; agreed success targets.
Connect Days 16–35	Establish a consistent starting point and keep source changes flowing.	Validated initial state, continuous capture, and quality results.
Publish Days 36–55	Put a governed product into a consumer's workflow.	Consumer using the product; freshness and business-response measures.
Operate Days 56–75	Confirm the product remains dependable under change and failure.	Recovery and verification results; service-level attainment and operating cost.



Phase	Business focus	Evidence for the next decision
Extend Days 76–90	Reuse the design for another consumer or supported destination.	Time and effort to extend; comparison with baseline; a funded next step.

The final review answers three questions. Did the first consumer get a better outcome? Did the product keep its operating commitment? Did the next use take less repeated work? The answers set the next product, the investment it needs, and which duplicate integrations can finally be retired.

Chapter 11. Your Next Step

11.1 Start Within a Clear Scope

Build the first engagement on the launch scope in Chapter 3.4, with source and target support confirmed for the chosen pattern [14]. Agree up front which runtimes Oracle manages and which deployments the customer operates, as Chapter 7 lays out. Keep the platforms and catalogs the customer already chose, run in parallel where a transition calls for it, and let verification make the cutover decision. When the team is ready to design, the Oracle GoldenGate Data Fabric product documentation has the deployment and support detail.

11.2 Strategic Conclusion and Your Next Step

For more than three decades GoldenGate has kept operational data moving while the infrastructure and analytical platforms around it were replaced, sometimes more than once and sometimes in reverse. GoldenGate Data Fabric extends that foundation into a shared experience for building, governing, and operating reusable data products. It connects current enterprise state to operational services, analytics, and AI, and it leaves the choice of where those workloads run with the customer. In a market full of platforms that optimize what happens inside their own boundaries, GoldenGate provides the real-time fabric between them.

Bring one costly delay, one planned platform transition, or one AI use case that needs data it can trust. Scope the 90-Day Change-to-Action Assessment with your Oracle team, set the baseline, and start a trial of GoldenGate Data Fabric on that one journey. Then expand on the evidence. Platforms will keep changing; that is the point. The fabric is what carries forward. Ninety days in, you will have your own numbers to decide with.



Appendices

Glossary and References

The glossary below contains terms that carry a specific meaning in this paper.

Term	Working definition
Operational-change-native data fabric	A data fabric that begins with committed enterprise transactions and business events, preserves operational context as change moves, and continuously produces governed data for operational, analytical, and AI consumption.
Operational truth layer	The business-facing description of the fabric: downstream analytics and AI grounded in authoritative enterprise data, committed change carried with transaction context, controls, provenance, and processing evidence. It does not mean independently determining real-world truth or validating model outputs (Chapters 3 and 5).
Dataset	The fabric's defining product abstraction: a durable logical lifecycle record for a governed data product, carrying transaction and CDC context, schema and endpoints, run evidence, quality and access controls, lineage, and physical-version history, independent of any one storage system, catalog, or engine (Chapters 3 and 7).
Data product	A governed, owned, contracted asset — rather than merely an output table. Includes owner, schema, quality rules, SLOs, policy, lineage, interface, and lifecycle. Chapter 7.3 describes where contract elements are recorded and where they are enforced.
Commit-to-action latency	The measurable path from a committed source transaction to an observed business action, decomposed into six stages (Chapter 1).
Data locality / data gravity	Placing pipeline execution beside the data it processes, because moving data-heavy workloads across clouds often costs more than the compute saved.
AI-ready data	Defined operationally: measurable freshness, completeness, consistency, schema compatibility, semantics, provenance, policy compliance, product SLO attainment, and authorized accessibility.



References

- [1] R. Thanaraj, A. Ronthal, E. Zaidi, and R. Sallam, "Evaluating Top 5 Trending Data Management Platforms," Gartner, ID G00859607, Jul. 16, 2026.
- [2] Oracle Product Development, "Oracle GoldenGate Data Fabric – AI Truth Layer for Distributed Data," Oracle AI World, Oct. 2026.
- [3] Oracle Product Development Blog, "Introducing Oracle GoldenGate Data Fabric," Oct. 23, 2026.
- [4] Oracle, "Oracle GoldenGate Platform: Innovations for Another 30 Years," [Online]. Available: <https://www.oracle.com/a/ocom/docs/middleware/data-integration/oracle-goldengate-innovations.pdf>
- [5] Oracle, "Oracle GoldenGate Advantages: Choosing the Right CDC Solution for Oracle Databases," technical paper, 2026.
- [6] Oracle, "How Oracle GoldenGate Veridata Works," Oracle GoldenGate 26 Documentation. [Online]. Available: <https://docs.oracle.com/en/database/goldengate/core/26/gvdug/how-oracle-goldengate-veridata-works.html>. Accessed: Jul. 15, 2026.
- [7] Oracle, "OCI GoldenGate Veridata Data Verification," Oracle Cloud Infrastructure Documentation. [Online]. Available: <https://docs.oracle.com/en/cloud/paas/goldengate-service/ocigg/about-data-verification.html>. Accessed: Jul. 15, 2026.
- [8] Oracle, "About OCI GoldenGate," Oracle Cloud Infrastructure Documentation. [Online]. Available: <https://docs.oracle.com/en-us/iaas/goldengate/doc/oracle-cloud-infrastructure-goldengate1.html>. Accessed: Jul. 15, 2026.
- [9] Oracle, "Regional Availability for Multicloud," Oracle Cloud Infrastructure Documentation. [Online]. Available: <https://docs.oracle.com/en-us/iaas/Content/multicloud/regions.htm>. Accessed: Jul. 15, 2026.
- [10] X. Gu and R. Thanaraj, "CODE: A New Data Management Platform Evaluation Framework to Fit Your AI Strategy," Gartner, ID G00854696, Jul. 1, 2026.
- [11] M. Miraz, H. Cook, R. Ramakrishnan, M. Gonzales, and A. Rosenbaum, "AI Success Requires Data Lakehouses; Accelerate Your Data Warehouse Transition Now," Gartner, ID G00846531, Feb. 24, 2026.
- [12] Oracle, "Oracle GoldenGate Microservices Architecture," Oracle GoldenGate 26ai Documentation. [Online]. Available: <https://docs.oracle.com/en/database/goldengate/core/26/coredoc/overview-oracle-goldengate-microservices-architecture.html>. Accessed: Jul. 15, 2026.
- [13] Oracle, "Securing OCI GoldenGate," Oracle Cloud Infrastructure Documentation. [Online]. Available: <https://docs.oracle.com/en-us/iaas/goldengate/doc/securing-oci-goldengate.html>. Accessed: Jul. 15, 2026.
- [14] Oracle, "GoldenGate Supported System Configurations." [Online]. Available: <https://www.oracle.com/integration/goldengate/certifications/>. Accessed: Jul. 15, 2026.
- [15] Oracle GoldenGate Data Fabric website, [Online]. Available: <https://www.oracle.com/goldengate/datafabric/>
- [16] Oracle, "MAA Platinum with Oracle GoldenGate," Oracle Database 19c High Availability Documentation. [Online]. Available: https://docs.oracle.com/en/database/oracle/oracle-database/19/haiaad/maa_platinum_ogg.html. Accessed: Jul. 15, 2026.
- [17] Oracle, "Use Oracle GoldenGate to Replicate Data with Autonomous Database," Oracle Autonomous Database Documentation. [Online]. Available: <https://docs.oracle.com/en-us/iaas/autonomous-database/doc/use-oracle-goldengate-replicate-data-autonomous-database.html>. Accessed: Jul. 15, 2026.
- [18] Oracle, "Cost Comparison of Equivalent Apache Spark Workloads Across Five Data Platforms," Oracle GoldenGate Research Document, Sep. 2026.
- [19] Oracle, "Oracle GoldenGate Product Overview." [Online]. Available: <https://www.oracle.com/integration/goldengate/>. Accessed: Jul. 15, 2026.
- [20] Oracle, "Using Oracle GoldenGate Studio 26ai: Pipelines and Recipes," Oracle GoldenGate Studio 26 Documentation, G49527-05. [Online]. Available: <https://docs.oracle.com/en/database/goldengate/goldengate-studio/26/ggsug/using-oracle-goldengate-studio.pdf>. Accessed: Sep. 13, 2026.
- [21] Oracle, "Mirror Data with ZeroETL Mirror Pipelines," Oracle Cloud Infrastructure GoldenGate Documentation. [Online]. Available: <https://docs.oracle.com/en/cloud/paas/goldengate-service/gzetl>. Accessed: Sep. 13, 2026.
- [22] Oracle, "Cascading," Oracle GoldenGate 26 Solutions Documentation. [Online]. Available: <https://docs.oracle.com/en/database/goldengate/core/26/ggsol/cascading.html>. Accessed: Sep. 13, 2026.



ORACLE CORPORATION

Worldwide Headquarters

Nashville, TN 37203 USA

Worldwide Inquiries

TELE + 1.650.506.7000 + 1.800.ORACLE1

FAX + 1.650.506.7200

oracle.com

CONNECT WITH US

Call +1.800.ORACLE1 or visit [oracle.com](https://www.oracle.com). Outside North America, find your local office at [oracle.com/contact](https://www.oracle.com/contact).

 blogs.oracle.com/oracle

 facebook.com/oracle

 x.com/oracle

Copyright © 2026, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 1026

October 2026
Author: Oracle Product Development



Oracle is committed to developing practices and products that help protect the environment