ACTIAN™
a division of HCLSoftware

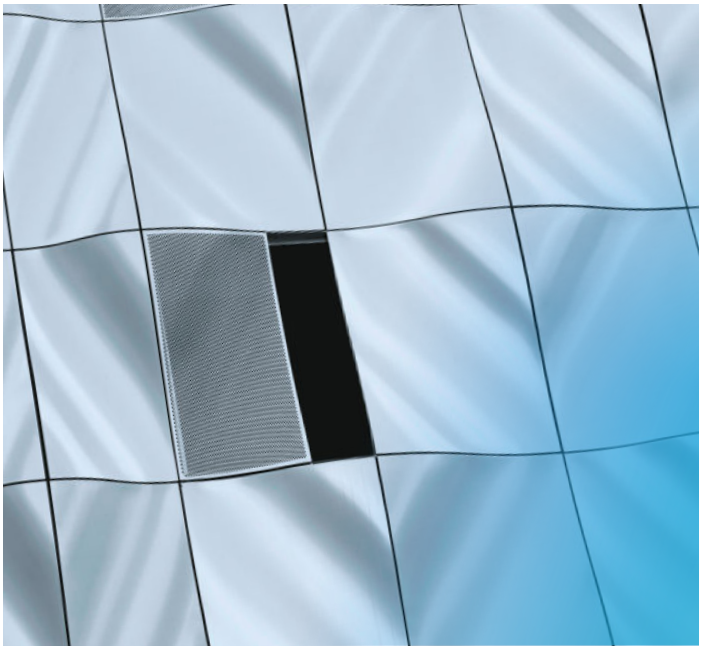
EBOOK

# Putting Data Contracts First

## The Engineer's Guide to Data Trust

## Table of Contents





## Engineering Trust

This guide will show you how to engineer trust directly into your data. We will introduce you to the **contract-first methodology**—an approach where reliability isn't an accident or a hard-won exception, but a guaranteed outcome. You will learn to build a world where data producers and consumers agree on the rules before a single byte of data is exchanged, creating an unbreakable bond of clarity and confidence from the very start.

If you're a data engineer, steward, or product manager, you know the daily reality of data anxiety. You're caught in a reactive loop of fire fighting quality issues, debugging broken reports, and answering the constant question: "Is this data right?" This endless cycle stems from an outdated paradigm where data is treated as a messy byproduct, not the valuable asset it is.

The solution isn't more dashboards or frantic cleaning scripts. It is a fundamental shift in approach: it's time to stop managing data chaos and start delivering data as a refined, reliable **product**.

In the following chapters, you will learn how to define, produce, govern, and consume data with a level of trust you previously thought impossible. Welcome to the future of data engineering.



ACTIAN™ a division of HCLSoftware

## Chapter 1: The Contract-First Philosophy: An Agreement Before an Asset

To build trust in data, we must begin with a new philosophy. The central idea is simple but profound: we must define the rules of engagement before the data is ever used. This is the essence of the **contract-first** approach. It's a commitment to creating an explicit, machine-readable agreement between a data producer and its consumers before the data product is even built.

### Why is the Contract-First Approach so Critical?

Because it forces the most important conversations to happen at the beginning of the process, not at the end when a dashboard is broken. When you define a contract first, you are no longer just documenting a table after the fact; you are co-designing a reliable asset with your stakeholders. Ambiguities about data meaning, quality expectations, and service levels are resolved from the outset, turning governance from a reactive, punitive process into a proactive, collaborative one. This philosophy is the architectural linchpin that makes trustworthy data at scale possible.

At the heart of this philosophy are two core concepts: the **data product** and its governing **data contract**.

### What is a Data Product?

A **data product** is not just a table in a database or a file in a data lake. It's a complete, packaged, and reusable data asset, designed to be valuable for a specific group of consumers. Think of it like any other product you might buy: it's well-documented, has a quality guarantee, and is easy to use.

A true data product has several key characteristics:

- **Discoverable:** It's easy to find in a central catalog or marketplace.
- **Trustworthy:** Its quality and reliability are guaranteed by the producer.
- **Self-Describing:** It comes with rich metadata and documentation that explains what it is and how to use it.
- **Interoperable:** It adheres to common standards that allow it to be easily combined with other data products.
- **Secure:** It has clear access control policies that are enforced automatically.

### The Anatomy of a Data Contract

If data is the product, then the **data contract** is its specification sheet, its user manual, and its quality guarantee, all rolled into one. The contract explicitly lays out the terms of the agreement, containing several essential components:

- **Schema:** The foundational blueprint defining the structure, such as column names, data types (e.g., `STRING`, `INT64`), and constraints (e.g., `NOT NULL`).
- **Semantics:** The layer of meaning that eliminates ambiguity. It clarifies business context, such as `revenue_usd` being post-tax or `event_timestamp` being in UTC.
- **Quality Metrics:** Specific, measurable rules about the data's integrity that go beyond simple data types. This includes guarantees about completeness ("`customer_id` must have 0% null values") or validity ("`country_code` must be a valid ISO code").
- **Service Level Objectives (SLOs):** Operational promises around the data's delivery, such as freshness ("updated every 60 minutes") or uptime.
- **Governance and Access:** The policies defining ownership, data classification (e.g., PII), and the roles permitted to use the data.

By establishing this complete agreement first, you are building on a foundation of clarity. The data product that follows is merely the fulfillment of this well-defined promise.

## Chapter 2: The Data Producer's Playbook: Engineering Trust at the Source

Theory is important, but seeing the **contract-first approach** in action is where its power becomes clear. Let's walk through a real-world scenario to understand how a data engineer can move from reactive firefighter to proactive trust builder.

### Use Case: Building a 'Golden Customer 360' Data Product

Imagine your company needs a single, authoritative source for customer information—a "**Golden Record**." Your task is to build this as a trusted data product.

1. **Define the Contract First:** Before writing a single line of code, you collaborate with the marketing and sales teams. Together, you draft a `contract.yaml` file. You agree on a schema (`customer_id`, `full_name`, `email`), a freshness SLO of 4 hours, and a critical quality rule: `customer_id` must be 100% unique and not null. This YAML file is checked into a Git repository.
2. **Implement the Pipeline with an API:** You now write the Spark or SQL code to ingest data from your CRM, e-commerce platform, and support system. As part of your CI/CD pipeline, you use a dedicated API to register and version your contract in a central data marketplace.
3. **Embed Automated Enforcement:** Your pipeline now has automated checks tied directly to the contract.
   - **At Build Time:** A junior engineer on your team tries to merge a code change that accidentally makes the `customer_id` column nullable. The CI pipeline fetches the active contract, sees that the change would violate the "`not null`" guarantee, and **automatically fails the build**. The breaking change is stopped before it ever reaches production, with a clear error message pointing to the contract violation.
   - **At Run Time:** Your pipeline includes a post-processing step using a tool like dbt or Great Expectations. If a source system change suddenly causes duplicate `customer_id`s, these tests—which directly reflect the contract's quality rules—fail the pipeline run. You are immediately alerted, and the corrupt data never populates the final data product.

The result? You have produced a "**Golden Customer 360**" that is both valuable and trustworthy. The contract is no longer just a document but an active, automated shield protecting the data's quality.

## The Principles that Made it Work

This use case was successful because it was built on two core principles of modern data engineering:

### Shifting Governance Left

"**Shift left**" means moving quality checks earlier in the development lifecycle. Instead of waiting for an analyst to find bad data, we used the data contract as an executable test suite to catch issues during development and data processing. We shifted governance from a downstream problem to an upstream solution.

### Automating Contracts via CI/CD

The process was not manual. By treating the contract as code (`contract.yaml`), it became part of the standard CI/CD workflow. It was version-controlled, and its rules were enforced automatically. This turns your CI/CD system—the engine of software development—into the engine of reliable data development.

By embedding these principles into your workflow, you move beyond hoping for good data and start engineering it.

# Chapter 3: The Data Consumer's Compass: Discovering and Activating Data Value

Trust is a two-way street. We've seen how data producers can engineer it, but it only creates value when data consumers can experience it. For data analysts, data scientists, and business users, a contract-first world transforms the painful process of finding and using data into a seamless, confident experience.

## Solving the Discovery Problem

In many organizations, finding the right data is a mix of tribal knowledge, Slack archeology, and hopeful guessing. A simple data catalog helps, but often it's just a list of tables with cryptic names and no context. A contract-driven data marketplace is different. It's less like a library card catalog and more like an app store for data—curated, searchable, and rich with context.

When a user searches for "customer data," they don't just get a list of 50 tables containing the word "customer." They are presented with certified data products. The "*Golden Customer 360*" product we built in the last chapter would appear at the top, clearly marked as "Certified," "High Quality," and with its purpose ("Authoritative source for all customer attributes") prominently displayed.

## The Data Contract as a 'Nutrition Label'

The data contract is the most powerful concept for a data consumer. It serves as a simple, easy-to-read "nutrition label" for every data product. Before running a single query, a user can instantly understand its fitness for their use case. When an analyst clicks on the "*Golden Customer 360*" product, they don't have to guess about its contents. They see:

- **Schema (Ingredients):** A clear list of all columns and their data types.
- **Freshness (Serving Size):** "Data updated every 4 hours."
- **Governance (Allergy Warnings):** "Contains PII. Not for use in public-facing dashboards."
- **Quality Guarantees:** "100% Unique customer_ids. Valid email formats."

This immediately answers the most common questions and builds confidence. The analyst knows exactly what they are getting. They can trust that this product is suitable for their critical marketing campaign analysis, without having to spend days validating its quality themselves.

## The Power of a Knowledge Graph

Modern data marketplaces powered by a knowledge graph don't just store a list of products; they understand the relationships between them. A consumer can visually explore:

- **Lineage:** "Show me exactly which source systems and transformations produced this '*Golden Customer 360*' product."
- **Downstream Impact:** "Which key dashboards and reports are powered by this data product?"
- **Related Products:** "What other certified products are frequently used alongside this one?"

This contextual discovery leads to faster, more accurate insights and prevents data misuse.

### Streamlined and Governed Access

The days of emailing a ticket to a faceless IT queue and waiting two weeks for data access are over. In a data marketplace, the process is integrated and transparent. A contract-first approach enables a far more sophisticated and automated method: **Attribute-Based Access Control (ABAC)**.

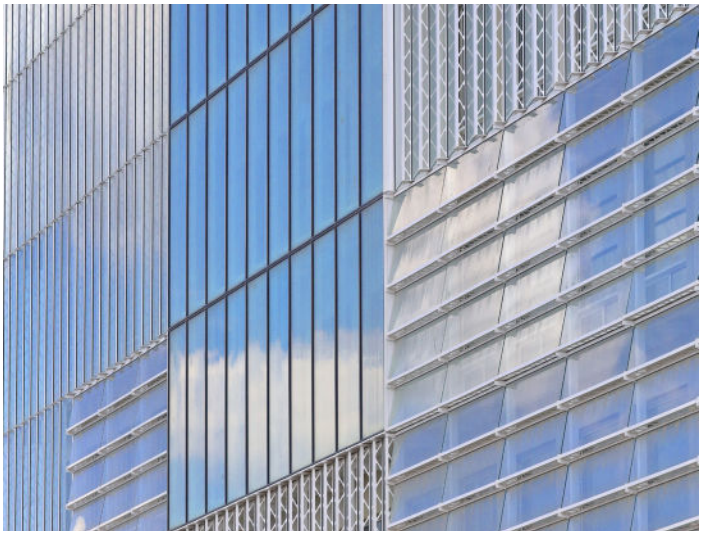
Instead of relying solely on static roles, ABAC uses the data contract's rich metadata to make real-time dynamic access decisions. Here's how it works:

1. **The Policy:** A central policy states, "A user can access a data product if their `department` attribute is 'Marketing' AND the data product's `classification` attribute is 'Confidential' OR 'Public'."
2. **The Request:** A marketing analyst finds the "*Golden Customer 360*" product. The system knows the analyst's department is "Marketing" from their user profile.
3. **The Contract:** The system reads the data product's contract and sees its `classification` is "PII/Confidential."
4. **The Decision:** The access control system evaluates the policy against the attributes:
   - User's department (`Marketing`) matches the policy.
   - Data's classification (`Confidential`) matches the policy.
   - Result: **Access is granted instantly and automatically.** No tickets, no waiting for manual approval.

If an engineer from the "Platform" team tried to access the same data, the system would see that their `department` attribute does not match the policy and automatically deny access.

The data contract is the engine for this process. Because it provides reliable, machine-readable attributes like `owner`, `classification`, and `domain`, the organization can build powerful, automated access policies. This moves beyond simple roles to a more granular, secure, and efficient system, perfectly balancing the need for speed with robust control.

For the data consumer, the entire journey—from discovery to analysis—is frictionless. They are empowered to self-serve with confidence, knowing that the data they are using is not just available, but certified, documented, and trustworthy.



## Chapter 4: Modern Governance in a Data Product World

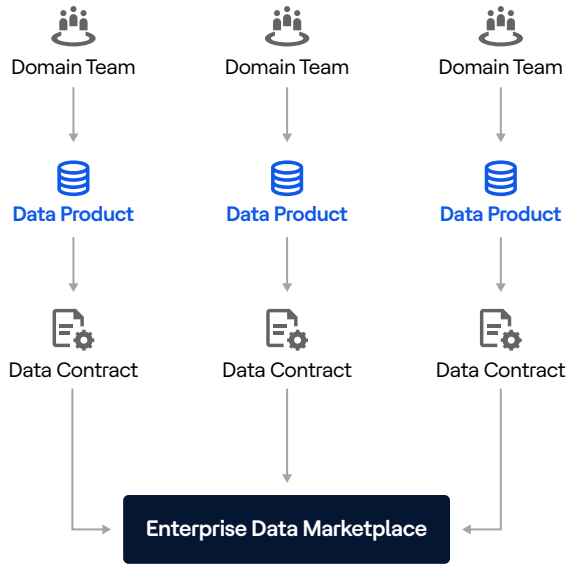
The shift to a contract-first model does more than just improve data pipelines and user experiences; it fundamentally reshapes the culture of data governance. It allows organizations to move away from a centralized, bottleneck-driven control model to a more federated, enabling one. This is where you evolve from being a data gatekeeper to a data enabler.

### Enabling Decentralized Ownership (Data Mesh)

The concept of a data mesh has gained enormous popularity because it addresses the scaling problems of monolithic data teams. A data mesh proposes that data should be owned and managed by the domain teams that know it best (e.g., the marketing team owns marketing data, the finance team owns financial data).

This model often fails in practice due to a lack of common standards. Without a shared way to ensure quality and interoperability, a data mesh can quickly devolve into a more chaotic version of data silos.

**Data contracts are the missing link that makes a data mesh work.** They provide the universal "shipping container" standard. Each domain can build its own data products in its own way, but to share a product on the enterprise marketplace, it must conform to a data contract. This provides the centralized governance and interoperability needed for the federated model to succeed.




## The Evolving Role of the Data Steward

In the old world, the data steward was often seen as a bad cop—chasing down teams to document their data, fix quality issues, and enforce rules. It was a thankless, manual, and often confrontational job.

In a contract-first world, the data steward's role is elevated to something far more strategic:

- **Marketplace Curator:** Instead of policing every table, the steward focuses on curating the enterprise data marketplace. They work to "certify" high-value data products, making them easy to find and promoting their use.
- **Standard Bearer:** The steward helps define the templates and standards for what makes a "good" data contract, ensuring consistency across the organization.
- **Data Literacy Champion:** They spend their time helping business users understand how to find, interpret, and use the certified data products available in the marketplace, rather than manually validating data for them.
- **Facilitator:** When new cross-domain data products are needed, the steward acts as a facilitator, bringing different domain teams together to agree on a shared contract.

## Data Lifecycle Management: Handling Change Without Breaking Things

Business needs change, and so must data. The question is how to update your data products without causing chaos for your downstream consumers. Data contracts provide a clear and safe process for managing this lifecycle. Here's how:

- **Versioning is Key:** Every change to a contract results in a new, semantic version (e.g., `dim_customer_v1.1`).
- **Non-Breaking Changes:** Adding a new, nullable column is a minor version change. Consumers are notified, but their existing queries won't break.
- **Breaking Changes:** Removing a column or changing its data type is a major version change (e.g., `dim_customer_v2.0`). This requires creating a new data product. The old version (v1.0) is marked for deprecation but kept running for a planned period. The marketplace automatically notifies all consumers of v1.0 about the deprecation schedule and points them to the new v2.0, giving them ample time to migrate.

This managed process provides the stability that businesses need while still allowing for the agility that data teams require

## Measuring Success

How do you know if your data product initiative is working? You can track a set of clear KPIs that demonstrate the shift from a cost center to a value creator:

- **Data Product Adoption:** How many certified data products are being used, and by how many unique consumers?
- **Time-to-Insight:** How long does it take for a new analyst to find the data they need and produce their first valuable report?
- **Reduction in Data-Related Support Tickets:** Are you seeing fewer tickets related to "bad data," "missing data," or "I can't find the data"?
- **Reuse Ratio:** Are new analytics projects being built by combining existing data products, rather than creating new data silos from scratch?

By tracking these metrics, you can prove the tangible business value of governing data as a product. You are no longer just managing a technical backend; you are cultivating a thriving ecosystem that directly fuels business success.

## Your First Step, Not Your Last: Activating Your Contract-First Strategy

You've reached the end of this guide, but you are at the very beginning of a transformative journey. You started with a problem you know all too well: a data ecosystem riddled with mistrust, inefficiency, and missed opportunities. Over these chapters, we have laid out a powerful, practical, and achievable blueprint for changing that reality. You now have the framework to transition from managing data chaos to engineering data trust.

### Recap: Your Journey to Governed Data

Now, let's recap the main points we've covered and identify the key takeaways that will form your action plan:

- **From Chapter 1, you learned the contract-first approach.** The key takeaway is the crucial paradigm shift: data is not a byproduct; it is a **product**. And every trustworthy product needs a guarantee. The **data contract** is that guarantee—a clear, enforceable agreement on structure, quality, and semantics that builds trust from the ground up.
- **From Chapter 2, you embraced the producer's playbook.** You saw how to **engineer trust at the source** by embedding data contracts directly into your CI/CD pipelines. The takeaway is that governance is not a downstream cleanup job; it is an automated, proactive part of the development lifecycle that prevents data issues before they ever occur.
- **From Chapter 3, you walked in the consumer's shoes.** You discovered how a contract-first approach transforms the user experience, turning data discovery from a frustrating treasure hunt into an intuitive exploration. The takeaway is that well-governed data **empowers your users**, giving them the confidence and frictionless access they need to drive real business value.
- **From Chapter 4, you adopted the mindset of a modern governor.** You learned that robust governance enables speed and autonomy; it doesn't stifle them. The takeaway is that your role can evolve from being a gatekeeper to an **enabler**, cultivating a thriving, trusted data ecosystem that fuels collaboration and innovation across the entire organization.

### Your Path Forward: From Reading to Doing

Knowledge is only powerful when applied. Here are three clear, actionable steps you can take—starting today—to turn these concepts into reality:

1. **Identify Your "Patient Zero."** You don't need to boil the ocean. Identify one critical data asset that is a constant source of pain, confusion, or mistrust. It could be a core dimension table, a key fact table, or a report that is always questioned. This is your pilot candidate.
2. **Draft Your First Data Contract.** Using the components outlined in Chapter 1, draft a v1 data contract for your chosen asset on a simple document or wiki page. Define the schema, write down the quality rules you wish it had, and state its intended freshness. Share this draft with one producer and one consumer of that data. This single act of creating clarity will spark a conversation that is more valuable than a dozen meetings.
3. **Advocate for an Automated Check.** Once the contract is drafted, work with your engineering team to implement just one automated check from that contract into your data pipeline. It could be a simple "not null" test or a check for valid enum values. When that check runs successfully for the first time, you will have officially begun your journey of shifting governance left.

This path is not an overnight transformation, but a series of deliberate, high-impact steps. **You now have the framework.**

## About Actian

Actian empowers enterprises to confidently manage and govern data at scale. Organizations trust Actian data management and data intelligence solutions to streamline complex data environments and accelerate the delivery of AI-ready data. Designed to be flexible, Actian solutions integrate seamlessly and perform reliably across on-premises, cloud and hybrid environments. Learn more about Actian, the data division of **HCLSoftware**, at **www.actian.com**.

Tel +1 512 231 6000 Fax +1.512.231.6010 710 Hesters Crossing Road, Suite 250, Round Rock, TX 78681 **Actian.com**



ACTIAN™
a division of HCLSoftware