



WHITEPAPER

The CDO's Guide to the Contextual Semantic Layer

How to build a governed business context foundation for trusted AI analytics

INSIDE THIS GUIDE

Contents

—	Executive introduction
01	The missing foundation beneath AI analytics
02	Model the business before generating metrics
03	Discover business knowledge and keep it current
04	Establish meaning before the question arrives
05	Certify the foundation in stages
06	Extend the foundation with semantic skills
07	A CDO roadmap for the first domain
08	Conclusion: The CDO's context mandate
—	How Codd AI applies this approach
—	Further reading

EXECUTIVE INTRODUCTION

Executive introduction

An executive asks, “How did gross margin change across our Western region over the last three quarters, and what drove the decline?” An AI assistant can write a query and return a chart in seconds. The speed is impressive. Whether the answer is useful depends on questions the assistant cannot settle from a prompt alone. Which business units belong to the Western region? Does gross margin include freight, rebates, and returns? Which date determines the quarter? Should intercompany transactions be excluded? Which product and customer relationships are valid?

Those answers exist somewhere in the enterprise. Some live in databases and transformation code. Others live in BI calculations, policy documents, spreadsheets, and the judgment of people who have run the business for years. If every assistant must reconstruct that knowledge when a question arrives, the enterprise can have a fluent interface without a dependable shared interpretation.

This guide explains how a contextual semantic layer can address that problem. It is a governed, reusable representation of business concepts, relationships, rules, vocabulary, and their links to data. It gives analytics tools and AI applications a common foundation from which to calculate metrics and interpret questions. AI can accelerate its creation. Business and technical experts must validate what the enterprise will treat as true.

For a Chief Data Officer, the opportunity is strategic. The same certified understanding can support dashboards, natural language analytics, data quality checks, and eventually more complex agentic workflows. The work is not complete when a model is generated. It requires ownership, change management, and evidence that answers remain consistent as the business changes.

01

SECTION 01

The missing foundation beneath AI analytics

Most enterprises have invested heavily in data infrastructure. They have warehouses, lakehouses, pipelines, catalogs, governance programs, and BI tools. They may also have several AI copilots. Yet a basic business question can still produce competing answers.

The reason is often a gap between data access and business understanding. A table can tell a system where a column resides and what type of value it contains. It rarely tells the system why one customer identifier should be used for sales analysis and another for billing, how a product hierarchy changed after an acquisition, or when a return should reduce recognized revenue.

A traditional semantic layer helps by defining joins, dimensions, measures, and metrics for analytical consumption. That work remains essential. AI raises the bar because users no longer choose from a small set of modeled dashboard interactions. They ask open-ended questions, combine concepts across domains, request explanations, and expect the system to decide which data and rules apply. A metric definition tells the system how to compute a value. It may not explain the business entities and relationships that make the computation appropriate.

Consider the gross margin question. An assistant needs to determine the requested measure, period, region, and grain. It must find the right transactions to include, apply the approved cost elements, and avoid duplicating rows when it joins product and order data. If the answer includes a recommendation, the system must also separate observed facts from interpretation.

The CDO's goal is to make that chain inspectable and repeatable. A contextual semantic layer stores the approved business understanding once, links it to physical data, and makes it available to different analytical experiences. It does not eliminate the need for source quality, sound analytical methods, or human judgment. But importantly, it gives them a common starting point.

1.1 A practical definition

A contextual semantic layer connects six things: business concepts such as customer and order; relationships such as an order belonging to a customer; vocabulary and synonyms used by different teams; rules and policies that govern interpretation; physical data models

that map concepts to sources; and certified business metrics derived from that foundation. It also records who approved important definitions and when they changed.

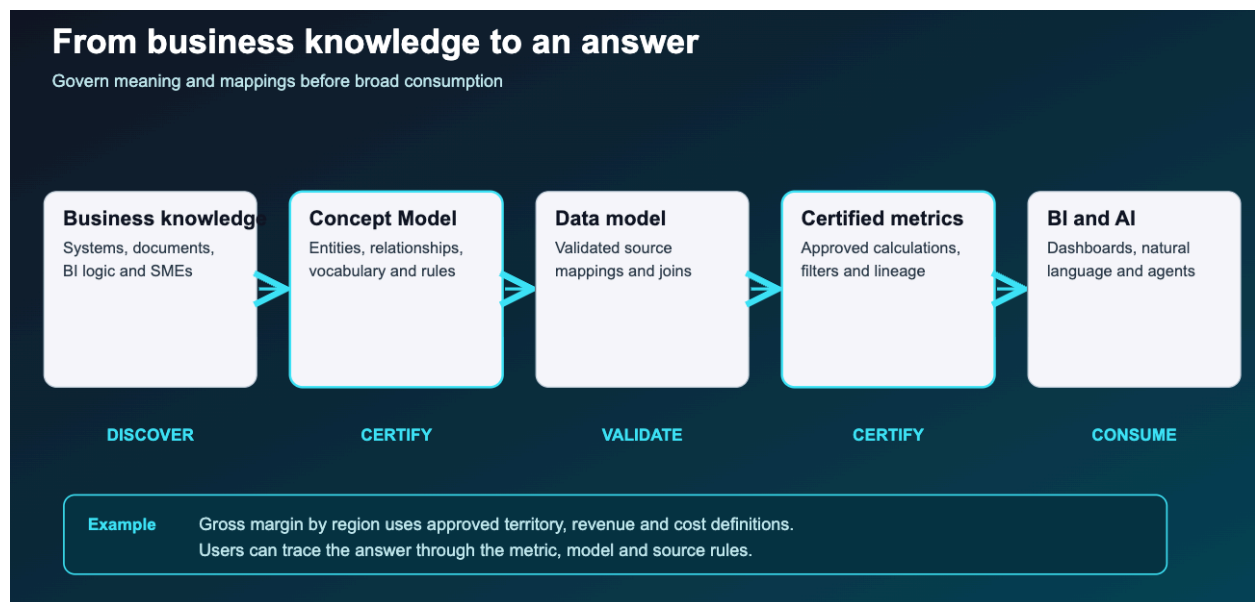
The layer sits between enterprise data and the applications that consume it. BI tools can use it to present consistent measures. AI assistants can use it to resolve a natural language question against approved concepts. Data teams can use the same relationships to identify data quality anomalies. The value comes from reusing governed meaning across those experiences.

02

SECTION 02

Model the business before generating metrics

An ontology is a structured representation of concepts and their relationships. In formal computer science, the term can refer to systems built with standards such as RDF and OWL and designed for logical inference. Codd AI uses the more specific term *Analytics Ontology* for a practical, machine-usable Concept Model built to ground analytics in an existing enterprise data environment. It is not a claim to build an OWL ontology or to run a formal logic reasoner.



That distinction matters for CDOs evaluating architectures. An analytics team may have hundreds of tables, incomplete documentation, and urgent requests. It needs a reliable model of the domain that links existing data to business meaning. It does not necessarily need the full machinery of a formal enterprise wide semantic model implementation. At the same time, a single metric or small list of metrics and SQL formulas are too narrow to represent the relationships an AI agent needs to navigate in a true decision support setting.

Start with the concepts behind the gross margin question. A customer buys a product through an order. An order contains lines. A product belongs to a category, and a transaction is attributed to a region according to an approved rule. Returns and discounts affect the recognized amount. Costs

may be associated with an item, shipment, or accounting period. The model should make those concepts and relationships explicit before anyone asks an AI system to assemble a margin analysis.

Then connect the conceptual model to physical data. The enterprise may have dozens of source systems, data marts or data lakes, but by focusing your project on a specific analytical domain narrows your focus to a set of database tables that you wish to include. The conceptual entities and relationships now need to be translated and connected to physical database tables and joins with the correct primary and foreign key assignments - along with the cardinality of each. A model that looks convincing at the concept level only can still produce wrong answers if its physical joins multiply rows or omit late-arriving transactions.

A key distinction in this process to the traditional BI semantic layer is that in addition to the technical database metadata, business documentation containing rules, logic and general knowhow is an explicit input to this process. The Concept Model and Physical Data Model is informed and enriched through the inclusion of the business knowledge to add descriptions, synonyms and other relevant information.

The business metrics come next. Once concepts, relationships, and rules are approved, the team can generate and certify measures such as net revenue, gross margin, gross margin percentage, and margin by product category. This reverses the repeated work of defining the same customer, region, and product logic inside every new metric. The Concept Model and Physical Model become a reusable source of meaning; metrics become governed expressions of it.

2.1 What the CDO should inspect

Ask whether the proposed model represents the language the business uses, whether each important relationship maps to data, and whether metric definitions reference approved concepts. Look closely at grain, effective dates, ambiguous terms, and exceptions. A diagram of entities is only a starting point. The test is whether a user can trace an answer from the question through the approved rule and relationship to the source records.

2.2 A worked example from question to answer

Take the request “Show gross margin for the Western region by product category.” The concept model identifies gross margin, region, and product category as approved business concepts. It resolves “Western region” to a defined territory hierarchy and determines whether the user wants the current hierarchy or the hierarchy in effect when the transactions occurred. The physical model identifies the approved sales and cost sources, their grains, and the join paths. The metric definition applies the approved treatment of returns, rebates, and costs. Access policy determines which business units should be included for the user to see.

The answer can then present a value and its basis: the period, definition version, filters, and source coverage. If the user asks why margin declined, the system can compare category contributions, price and cost movements, and the effect of returns. Those explanations are analytical interpretations. They should be distinguishable from the certified metric calculation and supported by the underlying comparisons.

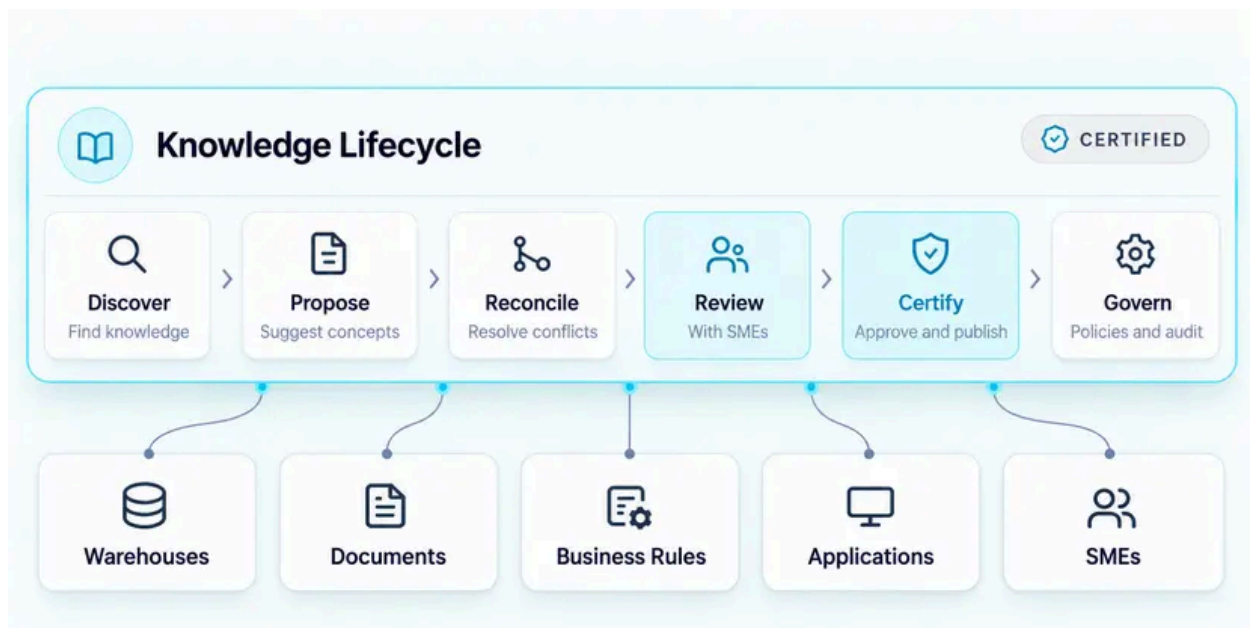
This example exposes a useful boundary. The contextual layer can establish what the measures and entities mean. It cannot guarantee that a model will choose the best causal analysis or that every source record is correct. Those capabilities require appropriate data quality controls, analytical methods, and review. A well-designed system makes each part visible instead of blending them into one confident sentence.

03

SECTION 03

Discover business knowledge and keep it current

In the previous section we proposed the explicit inclusion of business knowledge, rules and logic. A challenge for most organizations is that business context and knowledge are scattered across systems, reports, dashboards, file share locations and subject matter experts. Database schemas describe implementation. BI models contain calculations. Transformation pipelines encode filters. Policy documents explain intent. Subject matter experts know why a rule changed and where it has exceptions. Each source offers evidence, but no single source is authoritative for every question.



A practical discovery program starts with a defined analytical domain and a real decision objective. For gross margin, inventory the relevant data systems, transformation logic, reports, glossaries, finance policies, and business owners. Extract candidate concepts and rules. Record the source of each proposal, its effective date, and any conflict. Invite the people who own the process to resolve what the documentation cannot.

Suppose a BI report uses order date to assign a quarter while Finance uses posting date. Automated discovery should surface both definitions, not silently choose one. The steward and

Finance owner can decide whether they are different measures with different purposes or whether one implementation needs correction. The decision should be recorded with its scope and effective date so future users do not rediscover the disagreement.

The same discipline applies to apparently simple labels. “Western region” may refer to a current sales territory, a historical territory at the time of booking, or a finance reporting group. “Customer” may mean an account, a legal entity, or a billing relationship. The discovery process should capture synonyms and ambiguity, then identify which meaning applies to each analytical use case.

Discovery must repeat. New source systems arrive, territory boundaries shift, products are reorganized, and policies change. A productized process can rescan technical sources, detect changes in documents and BI logic, solicit SME review, and route proposed updates through certification. A change to a certified concept should identify dependent mappings, metrics, dashboards, and AI experiences that may need review. Without this feedback loop, a well-built foundation gradually becomes a historical account of the business.

3.1 A useful discovery record

For each important rule, capture the business statement, source evidence, proposed data mapping, owner, approval status, effective date, and dependent artifacts. Include unresolved conflicts. That record is more valuable than a glossary entry alone because it supports change management and answer lineage.

3.2 Resolve conflict without erasing legitimate differences

The purpose of governance is not to force every department to use an identical definition for every purpose. Sales bookings, invoiced revenue, and recognized revenue can all be legitimate measures. They should have distinct names, calculation rules, owners, and permitted uses. The problem arises when each is labeled simply “revenue” in a different tool and an AI assistant chooses among them without telling the user.

A review process can distinguish three cases. First, duplicate implementations of the same intended definition should be consolidated or reconciled. Second, genuinely different concepts should be modeled separately, with language that helps users choose. Third, a disputed definition should remain flagged until an accountable owner decides its scope. The semantic layer should preserve that decision and its effective date. It should not turn ambiguity into false certainty.

SME participation is most useful when experts see concrete proposals and exceptions. Asking a Finance leader to “review the ontology” is abstract. Asking whether freight belongs in gross margin

for a specific product line and period can produce a precise rule. Discovery technology should prepare that question, show the evidence, and make the answer reusable.

04

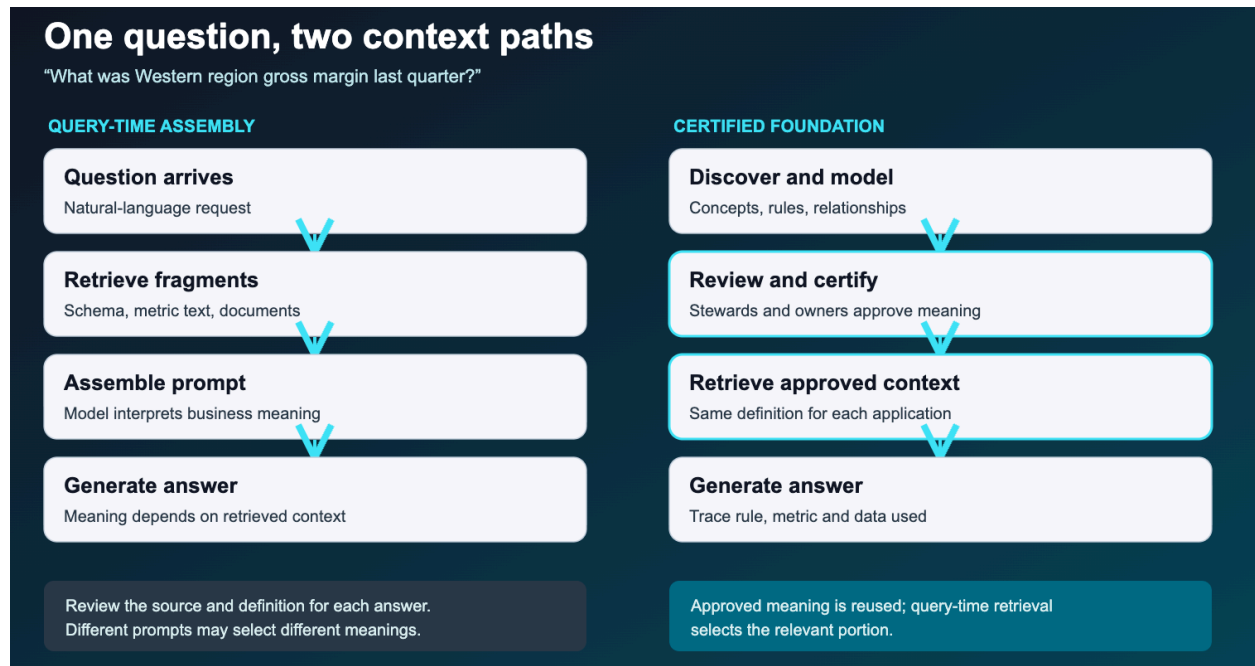
SECTION 04

Establish meaning before the question arrives

Many AI analytics experiences assemble context at runtime. A user asks a question. The application retrieves schema descriptions, perhaps a few metric definitions or documents, adds them to a prompt, and asks a model to produce a query or answer. This pattern can be useful. Retrieval helps select relevant material, and a model can reason over the current question. The risk appears when the application must also decide the business meaning of a term from incomplete, inconsistent, or unapproved material every time.

Imagine two employees asking for Western region gross margin. One assistant receives a prompt with a Finance definition. Another retrieves a sales dashboard description. A third finds only column metadata. All three may produce plausible charts. Even if each query runs correctly, the answers may differ because the underlying meanings differ. The problem is difficult to detect when the output looks polished.

Metric-level documentation helps, but it can create its own maintenance burden if every new measure restates the same relationships and rules. When a region mapping changes, teams must locate all copies of that logic. If they miss one, the organization develops semantic drift: analyses that once agreed begin to diverge.



A governed contextual foundation changes the sequence. The enterprise identifies and certifies concepts, relationships, mappings, and rules upfront before broad consumption. At query time, an assistant can still retrieve the relevant subset and reason about the user's request. It draws from approved context with known ownership and lineage. Runtime context remains a delivery mechanism; it is no longer the sole place where business meaning is invented.

This is an architectural choice, not a promise of identical prose or flawless reasoning. Two valid analyses may answer different questions, and a generative model may phrase the same result differently. The goal is to make the data selection and business definitions consistent, and to expose the assumptions behind analytical interpretation.

4.1 Five questions for an architecture review

Where is the approved definition of each core business concept? How is it linked to physical sources? Who can change it? Which applications consume the same definition? Can an answer show which version of a rule it used? If the team cannot answer these questions, adding more prompt instructions will not by itself establish durable governance.

4.2 Compare approaches by their operating burden

An architecture review should go beyond a vendor feature checklist. Ask a team to make a realistic change, such as moving three states into a new sales region or redefining the treatment of returns. Count the places where the rule must be edited. Identify the dashboards, metrics, prompts, and

agents affected. Check whether the system detects dependencies, requests the right approvals, and shows which answers used the earlier version.

Run the same representative question through two consuming applications. If the applications return different numbers, determine whether the difference reflects permissions, timing, a legitimate alternative definition, or duplicated logic. This exercise reveals the maintenance cost of context more clearly than a demonstration of a single natural language query.

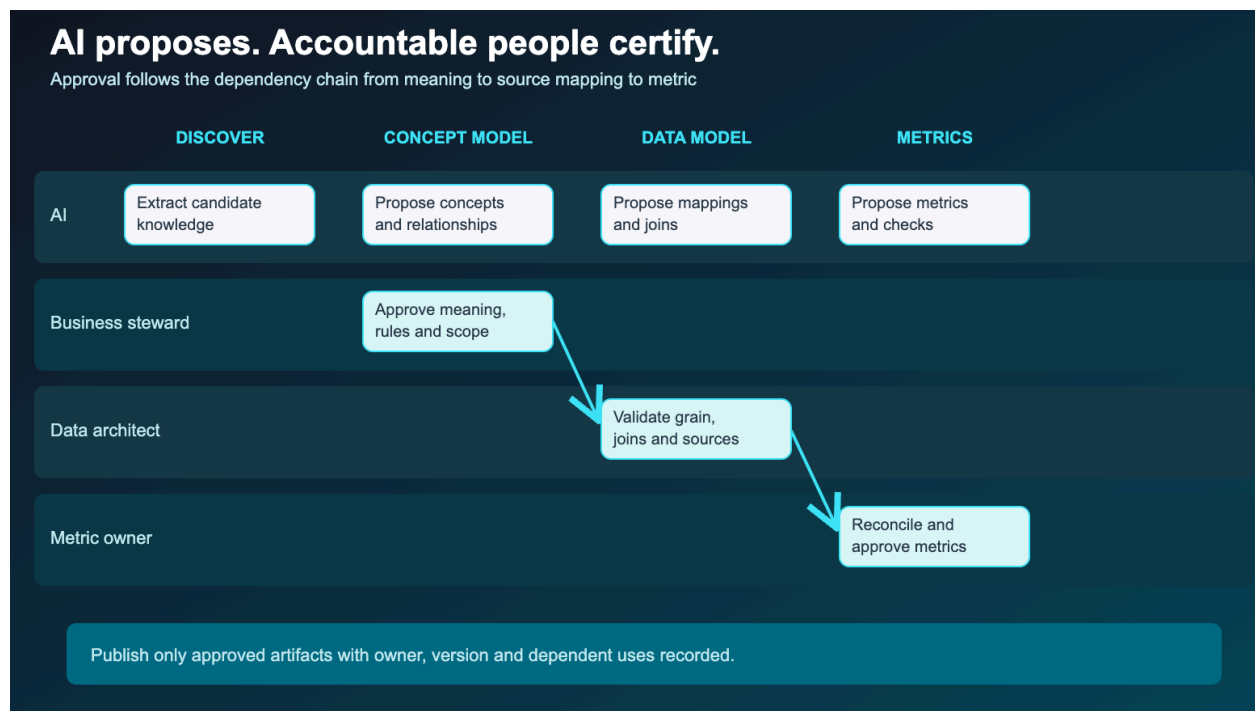
Also test failure. Ask for a metric that has no certified definition and for a combination of entities that lacks a validated relationship. A trustworthy experience should expose the limitation or route it for review. It should not invent a join merely because a plausible column name exists.

05

SECTION 05

Certify the foundation in stages

AI can accelerate semantic engineering by analyzing schemas, documentation, samples, and existing calculations in BI reports and dashboards. It can propose concepts, relationships, synonyms, mappings, and metrics. Proposals are valuable because they reduce the blank-page work. They are not automatically enterprise policy.



Certification should follow the dependency chain. First, business owners and data stewards review the proposed concepts and vocabulary. Are “booked revenue” and “recognized revenue” distinct? Does a customer group have a specific legal meaning? Next, architects and engineers validate the physical model, join paths, grain, and source coverage. Then metric owners review formulas, filters, dimensions, exceptions, and reconciliation to accepted reports. Only approved artifacts should become the foundation for broad analytical consumption.

The review needs evidence. For each important concept, show where it came from and which owner accepted it. For a relationship, show the proposed mapping and test it against representative records, including exceptions. For a metric, compare the calculated result with a trusted baseline

over a defined period and investigate any difference. Record approval, version, and dependencies so a later change can trigger the right reviews.

Human review should be concentrated where it changes the outcome. A steward need not inspect every synonym with equal effort. Material revenue rules, ambiguous joins, policy constraints, and high-impact metrics deserve deeper scrutiny. Confidence scores can direct attention, but a high score is not a substitute for business ownership.

Governance also continues after release. Monitor questions that could not be resolved, corrections from users, changes in sources, and divergent answers. Route those signals back into discovery and certification. A foundation earns trust through an operating process, not a one-time sign-off.

5.1 Roles that make certification work

The business owner defines the meaning and acceptable use of a concept. The data steward manages definitions, evidence, and change requests. The data architect validates structure, joins, and dependencies. The metric owner confirms calculations and reconciliation. Security and governance teams define access and policy constraints. A CDO can assign these responsibilities in an existing governance model rather than create an entirely new committee.

5.2 Treat approval as a versioned decision

Certification is most useful when it is specific. A steward might approve the concept “active customer” for a particular domain while withholding approval of a proposed mapping to a new CRM system. A metric owner might approve monthly gross margin after reconciliation but leave a daily version under review because costs arrive late. The system should make those distinctions visible to consumers.

When a rule changes, preserve the earlier approved version where historical reproducibility matters. Record the reason for change, effective date, approving owner, and affected outputs. This lets a team explain why a report issued last quarter differs from a current restatement. It also helps an AI application answer a question about a past period without applying today's hierarchy to yesterday's organization by default.

Access governance belongs in the same operating design. A shared definition does not mean unrestricted access to all underlying records. The consumption layer must honor the user's entitlements and distinguish a limited view from an enterprise total. Consistent meaning and appropriate access are complementary requirements.

06

SECTION 06

Extend the foundation with semantic skills

The first output of a contextual semantic layer is often a set of business metrics. From approved customer, product, order, and regional relationships, a team can derive revenue by product family, gross margin by region, order fulfillment performance, and other measures. Users can consume them in dashboards or natural language analytics.

The same certified relationships can serve a different purpose. Consider a data quality objective: identify orders that refer to customers that do not exist, products without approved suppliers, or records missing mandatory attributes. A data quality skill applies the shared model to generate candidate checks and monitoring metrics. The business context stays consistent; the analytical objective changes.

This matters because quality rules often have business meaning. A missing value is not always an error. A supplier may be optional for one product class and mandatory for another. A zero-dollar order may be an approved sample or a broken transaction. The concept model and business rules help distinguish a meaningful exception from a generic null check. Humans still review and certify proposed checks, thresholds, and remediation paths.

The data team might then monitor unresolved customer references, affected orders, and the age of each exception. A business leader might see whether unreliable records could change a regional margin report. Both views use the same underlying understanding but answer different operational questions.

Once a quality issue is confirmed, a defined workflow can identify the owner, open a ticket, assess affected reports, and notify the appropriate team. The system should make its evidence and proposed action visible. Higher-impact actions require the approval and access controls appropriate to the organization. Other semantic skills, such as governance or compliance checks, are possible extensions; each needs its own validation and operating controls before it is trusted.

The economic point is straightforward. When a new use case can reuse certified concepts and mappings, teams spend less time reconstructing business context. The return depends on how well the foundation is maintained and how much genuinely reusable knowledge it contains.

6.1 A second example for the data team

Suppose the business relationship says that every order must refer to a valid customer, except for a defined class of anonymous retail transactions. A generic database check for a non-null customer ID would miss invalid references and wrongly flag permitted anonymous transactions. A data quality skill can propose a more meaningful rule from the certified relationship and its exception. The data steward and order-process owner validate the rule before it is monitored.

The resulting measure is “orders with an invalid customer relationship,” with a clear denominator, exception handling, and owner. A dashboard shows the trend and affected systems. An operational workflow may notify the responsible team after a threshold is crossed. If an AI agent suggests remediation, it should show the records and rule version supporting its suggestion. The same concept model that supports customer revenue analysis now helps the data team protect the quality of that analysis.

The distinction between a skill and an autonomous action is important. Generating a candidate check, certifying a check, detecting a breach, and changing source data are separate steps with different permissions and review needs. A CDO can extend capabilities in stages, preserving the speed of discovery without granting an agent authority beyond the organization's controls.

07

SECTION 07

A CDO roadmap for the first domain

Begin with one business decision where inconsistent answers have a visible cost. Financial or sales analysis are good options. Gross margin analysis, customer retention, inventory availability, or order fulfillment may be good targets to your organization. Choose a domain with an engaged business owner, accessible and ready source data, and a few existing reports against which results can be checked.

PHASE

1

Phase 1: Define the analytical objective and baseline. Write three to five representative questions, including an ambiguous one and a follow-up that crosses data domains. Collect the accepted answers or reconciliation sources. Record current time to answer, manual effort, and known definition disputes. Avoid declaring success solely because a chatbot can produce a chart.

PHASE

2

Phase 2: Discover and resolve context and develop physical data model. Inventory technical metadata, transformations, BI calculations, policies, glossaries, and SME knowledge. Build a proposed concept model and log conflicting definitions. Assign an owner to each consequential decision. This work establishes what the enterprise means, not merely what the current application happens to calculate. Once the concept model is certified generate the physical data model and how it relates to the concept model. Again, assign data engineers and architects to validate and certify the model.

PHASE

3

Phase 3: Generate the first 25 to 30 high value metrics for your use case. Validate and certify each of the metrics for accuracy of results and the underlying SQL or YAML definitions. Reconcile them over representative periods and edge cases. Where two legitimate definitions exist, label their purpose and scope rather than forcing an artificial single answer.

PHASE

4

Phase 4: Connect one consumption experience, ideally a natural language interface that will allow AI reasoning and deterministic question responses to be observed and verified. Test the original questions with different phrasing and different users. Review data lineage, rule selection, access controls, and the presentation of uncertainty. Capture corrections as structured feedback.

PHASE

5

Phase 5: Expand carefully. Add adjacent concepts and additional consumption interfaces such as BI tools, dashboards or agentic workflows. Set triggers for recertification when source schemas, policies, hierarchies, or business owners change. Track reuse: how many experiences consume the approved concepts, how often definitions are duplicated elsewhere, and how quickly a rule change propagates.

7.1 How to judge progress

Measure answer agreement against certified definitions, the proportion of priority concepts with clear owners, reconciliation exceptions, time required to approve a new metric, and the time between a business change and an updated certified model. Another useful process to track is the frequency of requests from users for approval of new metrics generated by the system upon end user requests. These are great to measure user engagement but also keep building the foundational corpus that will make all AI more robust and trusted. These are operating measures, not universal benchmarks. Establish a baseline for the chosen domain and improve from there.

7.2 Make the first review concrete

At the end of the initial domain deployment, ask the business owner to review the original questions and several unplanned variations. Ask an analyst to trace each answer to the metric and source. Ask an architect to inspect join grain and exceptions. Ask a steward to propose a definition change and follow its approvals and downstream effects. Ask a user with narrower permissions to run the same query. Finally, ask a question the foundation does not yet support.

The exercise should yield a short list of defects and decisions. Some defects will be data quality issues; some will be missing knowledge; some will be weaknesses in the AI application's reasoning. Classify them accurately. The contextual layer can make business meaning governable, but it should not be credited with solving every adjacent problem or blamed for every model-generated mistake.

Expansion should follow observed reuse. If the first domain's concepts are frequently needed by adjacent teams, connect those teams to the existing foundation and resolve boundary terms such as “customer” across domains. If the next use case requires entirely different business concepts, budget for its discovery and certification. A shared platform reduces duplication; it does not make domain expertise free.

08

CONCLUSION

The CDO's context mandate

AI makes it easier for people to ask questions of enterprise data. It also makes unresolved business definitions visible at a scale that dashboards rarely did. The CDO can address the underlying issue by treating business context as a governed asset: discovered from real sources and experts, modeled as concepts and relationships, connected to physical data, certified by accountable owners, and refreshed as the business changes.

The practical test is the executive's gross margin question. Can the organization explain what the question meant, which data and rules produced the result, who approved those rules, and what would change if a policy or source system changed? If so, it has a foundation that can support many AI experiences. If each application answers those questions independently, every new interface can recreate the same uncertainty.

Start with one domain and analytical insights that matters to the business. Build the shared understanding, certify it, and prove that it can be reused. Then extend its reach to new metrics, quality checks, and analytical workflows without rebuilding the meaning of the business each time.

8.1 CDO discussion checklist

- 01 ☐ Which business questions most often produce conflicting answers?
- 02 ☐ Where do the approved definitions and exceptions for those questions live?
- 03 ☐ Can we identify the owner and evidence behind each important rule?
- 04 ☐ Do our concept relationships map to tested data joins at the correct grain?
- 05 ☐ Which metrics repeat logic that should be governed once?
- 06 ☐ Can BI tools and AI applications consume the same certified definitions?
- 07 ☐ Can a user trace an answer to its data, rule version, and metric definition?
- 08 ☐ What triggers review when a source system, policy, or hierarchy changes?

- 09 ☐ How do users report an ambiguous or incorrect interpretation?
- 10 ☐ Which second use case could reuse the foundation we build first?

ABOUT CODD AI

How Codd AI applies this approach

Codd AI is designed to help organizations build and operate a contextual semantic foundation for analytics. It brings together technical metadata and business knowledge, uses AI to propose an analytical Concept Model, physical data model, and metrics, and keeps business and technical experts involved in reviewing and certifying those artifacts. The certified foundation can then support conversational analytics, BI, and other AI-driven experiences.

The approach starts with business understanding. Its value grows when organizations keep discovering and validating knowledge as systems and policies evolve. The same certified foundation can support different semantic skills, including business metric generation and data quality analysis, while preserving ownership and review for each new use.

To explore how this approach could apply to one of your domains, visit www.codd.ai.

RESOURCES

Further reading

Beyond Business Metrics: Teaching Your Semantic Layer New Skills.

<https://www.codd.ai/blog/beyond-business-metrics-teaching-your-semantic-layer-new-skills>

The Contextual Semantic Layer - Powering Trusted GenAI Analytics.

<https://www.codd.ai/blog/contextual-semantic-layer-powering-trusted-genai-analytics>

What Ontology Actually Means for Analytics.

<https://www.codd.ai/blog/ontologies-market-confusion-analytics-ontology>

Why Runtime Context Is the Wrong Foundation for Enterprise AI Analytics.

<https://www.codd.ai/blog/runtime-context-wrong-foundation-enterprise-ai-analytics>