EXADEL AI

The Agentic AI Data Architecture Playbook

A concise guide to designing, building, and governing modern data architecture for AI agents

info@exadel.com

2026

Confidential

Agentic AI needs a different kind of data foundation

When agentic AI systems fail, the primary cause isn't weak models. It's fragmented, poorly governed data that isn't shaped for autonomous consumption.

An agent working from incomplete or unreliable data can make confident but wrong decisions at machine speed. And when that agent works across enterprise systems, those decisions can create serious operational and governance risks. To reduce those risks, you need to modernize your data stack around how agents retrieve, interpret, and act on data. In practice, that means five architectural changes:

- **Context travels with governance.**
  Data carries its lineage, quality signals, and access policies through the pipeline.
- **Meaning is explicit.**
  Ontologies and knowledge graphs give agents a shared understanding of business information and how it relates.
- **Unstructured data becomes first-class.**
  Documents, conversations, images, and other unstructured content become reliably retrievable through vectorization and embedding services.
- **Data is packaged for reuse.**
  Owned data products with defined interfaces, SLAs, and semantics give agents trustworthy assets instead of requiring a new pipeline for every use case.
- **Access is governed at the point of use.**
  Each agent has its own identity, so access controls and an AI gateway can enforce what data it can access and what actions it can take.

The result is an architecture in which agents can access the context they need while the enterprise retains control over the data, policies, and decisions behind every interaction.

Five layers. One governed architecture.

The architecture follows data from the moment it enters the enterprise to the point where an agent retrieves it, interprets it, and uses it to act. Each layer prepares the data for the next. Governance, identity, lineage, and observability run across the entire architecture.

Layer 5 **Data Consumption Layer**

At the top of the architecture, trusted data and intelligence enter real workflows. Orchestration coordinates agent activity, retrieval services supply the right context, and the AI gateway controls how agents and models access enterprise resources. Agent outputs can also flow back into the architecture as data, creating a feedback loop.

Layer 4 **Data Products Layer**

That trusted, semantically consistent data can then be packaged for reuse. Data products bring together curated tables, graph views, embeddings, and metrics as owned business assets with defined interfaces, contracts, SLAs, semantics, quality expectations, and access controls. Agents can reuse them rather than rebuilding the same data logic for every task.

Layer 3 **Semantic Layer**

Making data retrievable is not enough. Agents also need to understand what it means. The Semantic Layer uses ontologies and knowledge graphs to define business entities, relationships, and rules, giving different agents a shared, machine-readable understanding of the business.

Layer 2 **Data Platform Layer**

The platform turns that incoming data into something agents can reliably retrieve and use. Data is stored, transformed, and organized through the lakehouse, while unstructured content is vectorized for meaning-based retrieval. The platform also gives agents a controlled way to work across systems, retain the context they need, and access data according to their identity.

Layer 1 **Data Source Layer**

It all starts with the enterprise's existing data. Structured and unstructured data enters from operational systems, applications, documents, conversations, and other sources. Rather than leaving governance until later, quality controls, PII handling, lineage, and other policies are applied as the data is captured and prepared.

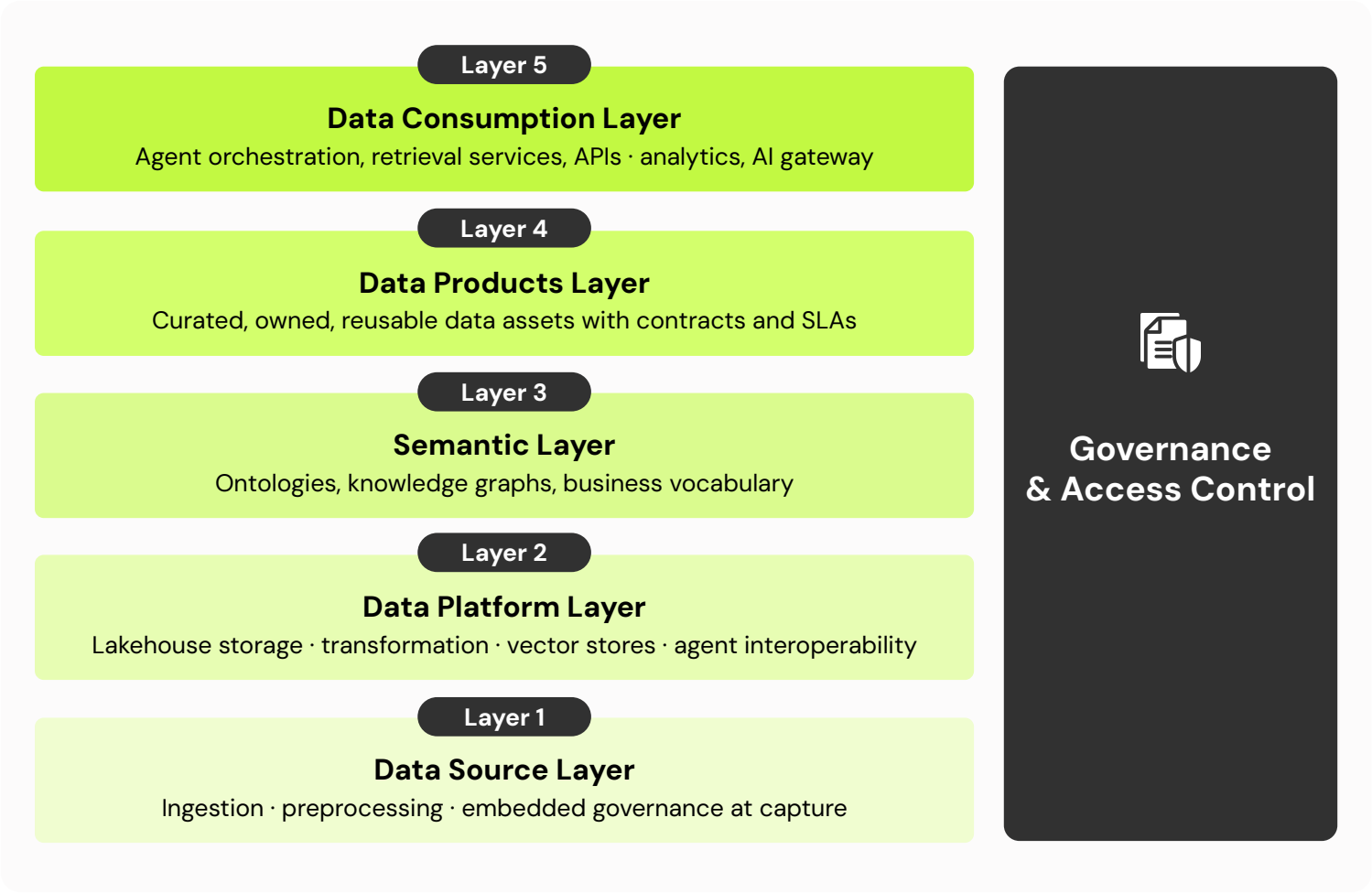
Governance & Access Control

Governance is not a sixth step after consumption. It runs through all five layers. Access controls determine what each agent can use, while lineage and monitoring provide a record of what it did.

Modernization doesn't mean replacement

Each layer maps to systems most enterprises already operate: lakehouses, ETL/ELT tools, data catalogs, APIs, and other existing platform capabilities.

The agentic shift is to build on that foundation with vectorization, semantic modeling, agent identity, dynamic orchestration, and automated governance.



Layers 1 & 2

From raw data to agent-ready data

Agents need access to structured transactions, event streams, documents, conversations, media, and third-party data. Layers 1 and 2 provide the foundation that gets this information into the architecture, governs it, transforms it, and makes it retrievable.

Layer 1: Data Source Layer

The Data Source Layer is where raw enterprise data enters the architecture.

For agentic AI, the critical change is that governance must begin as the data arrives. Quality checks, PII detection, and lineage can't wait for downstream review. By then, the data may already have been transformed, embedded, or combined with other sources. Different sources require different ingestion patterns. Transactional systems, event streams, reference data, documents, and media all enter differently, but the same principle applies: governance begins at capture.

Whatever the source, these four controls should be embedded at capture:

- **Schema and data contracts** flag records that violate agreed structures.
- **Automated quality checks** test incoming data as part of the pipeline.
- **PII and sensitive-data scanning** tags regulated or personal information so downstream policies can act on it.
- **Lineage capture** records where data originated and how it moves through the architecture.

Before the data moves further through the architecture, preprocessing cleans, enriches, labels, and scores it so later layers have more reliable information to work with.

Layer 2: Data Platform Layer

The Data Platform Layer connects data across systems and makes it usable by applications and agents.

It has four core responsibilities: storing and transforming data, making unstructured content retrievable, helping agents work across systems, and controlling access. A lakehouse provides the storage and transformation foundation. Raw data is progressively cleaned, standardized, and curated for business use. An additional agent-ready tier prepares that data for agent retrieval with semantic tags, access controls, and links to relevant embeddings.

Making unstructured data retrievable

Vectorization is one of the capabilities that most clearly distinguishes an agent-ready architecture from a traditional analytics stack. Documents, transcripts, product descriptions, images, and other unstructured content are transformed into embeddings so information can be retrieved based on meaning rather than keyword matches alone.

The process includes extraction, chunking, metadata attachment, embedding generation, indexing, and a defined freshness or re-embedding strategy. In production, retrieval should combine vector similarity with keyword matching and metadata filtering rather than relying on pure vector search.

Every agent also needs its own service identity. Access to shared data can then be evaluated according to the requesting agent, its task, and current policy, while actions remain attributable and auditable.

Layers 3 & 4

From shared business meaning to reusable data

Making information retrievable isn't enough. Agents also need to understand what that information means and use it consistently. Layers 3 and 4 are where governed data gets turned into shared business knowledge and packaged into reusable assets.

Layer 3: Semantic Layer

The Semantic Layer defines what business information means, how it relates, and what rules apply to it in a form both people and machines can understand. Without this layer, agents could interpret the same underlying data differently depending on the table, system, or document they retrieve.

Two components make this possible.

Ontologies define business reality.
They establish the entity types that exist in a domain, their attributes, the relationships between them, and the rules or constraints that apply.

Knowledge graphs apply that ontology to real enterprise data.
Actual customers, orders, products, and interactions become connected entities drawn from the systems ingested and prepared in the preceding layers.

Agents can access this shared meaning through governed services such as GraphRAG and semantic query APIs. Rather than retrieving disconnected pieces of information and trying to reconcile them, an agent can retrieve connected context based on the relationships between entities.

This shared vocabulary also helps keep agent responses grounded by limiting them to recognized business terms and values.

Layer 4: Data Products Layer

The Data Products Layer packages curated tables, graph views, embeddings, or metrics into reusable, business-ready assets.

Each data product has:

- a named **owner**
- a defined **interface**
- a **contract and SLA**
- agreed **semantics**
- monitored **quality metrics**
- an **access policy**

Instead of creating different versions of the same data, agents and applications can use the same trusted asset. Defined interfaces make its use observable, while contracts and deprecation policies help prevent downstream agents from breaking silently when the underlying data changes.

The progression is deliberate: the Semantic Layer establishes shared meaning. The Data Products Layer makes that meaning reusable.

Layer 5

Putting trusted data to work

The Data Consumption Layer

This is where data and intelligence enter workflows and applications. For agentic AI, it's where orchestration, retrieval services, and the AI gateway operate. Unlike a traditional application working from predefined queries, an agent decides at runtime what context it needs. It retrieves that context, evaluates whether it is sufficient, and refines its retrieval when needed.

The orchestration layer makes this possible through:

Dynamic context assembly

Structured data, semantic relationships, and vector retrieval can be combined into the context an agent needs.

Tool and function routing

Governed interfaces connect agents to the data products and platform services they are allowed to use.

Iteration control

Retrieval and refinement operate within defined cost and latency limits, with fallback behavior when information is insufficient.

Retrieval services find the most relevant context for an agent, combining semantic and vector search where needed. Access policies are applied before results reach the agent, and retrieved information can be re-ranked for relevance. Retrieval gives the agent the context it needs. The AI gateway controls the interaction between agents, models, data, and tools. It handles:

- authentication and authorization for individual agent identities
- usage policies and permitted actions
- prompt and response logging
- PII and sensitive-content filtering

Agent outputs can also become data. That means generated labels, resolved intents, and successful resolution patterns can flow back into the platform under the same governance controls and improve future retrieval.

Governance & Access Control

Governance isn't the final stage, but a vertical control plane spanning the entire architecture.

It works in two ways: data is progressively curated while its lineage is preserved, and identity-driven access controls are enforced when data is requested and used.

Every human, application, and agent has a distinct identity. Access decisions are made based on that identity, the role assigned, data sensitivity, requested action, and task context. Agents start out with least-privilege access, and every access decision is logged.

Recording which data products agents use and what information they retrieve makes their behavior traceable. It also provides feedback for improving retrieval, embedding freshness, and semantic coverage over time.

One customer journey. Five connected layers.

Consider a customer who views a jacket in a mobile app and adds it to their wishlist. They later place an order, but it's delayed in transit. They call support about the delay and, later that day, ask the same question through webchat.

They expect the company to already know what happened.

1 Data Source Layer

The wishlist event, order, support call, and webchat message enter the architecture through the appropriate ingestion paths. PII is handled, schemas are enforced, and lineage is captured.

2 Data Platform Layer

Order and delivery information becomes available for real-time access. The support transcript is processed, chunked, embedded, and indexed with relevant metadata. The webchat agent has its own authenticated identity and scoped access.

3 Semantic Layer

The knowledge graph connects the customer to the wishlist item, order, delivery event, and support interaction. These separate records now form a connected business context rather than isolated events.

4 Data Products Layer

The webchat agent uses a governed Customer 360 data product to retrieve the customer's profile, order, and interaction history through one versioned interface rather than querying raw tables.

5 Data Consumption Layer

Orchestration assembles the context. A graph traversal identifies the delayed order and recent support interaction. Retrieval brings in the relevant part of the call transcript. The AI gateway checks the agent's permissions and logs the prompt and supporting context.

The agent can now respond with awareness of the earlier call. The customer doesn't need to explain the problem all over again.

Governance across the journey

PII handling, lineage, entity resolution, access decisions, and prompt activity are recorded throughout. If the interaction later needs to be reviewed, the path from source data to agent response is auditable.

The value comes from the architecture working as a whole

Without this connected architecture, the webchat agent may have no way to know that the customer called only minutes earlier. The customer has to repeat the problem because the context from one interaction hasn't followed them into the next.

The payoff isn't any single layer. It is making a customer action available, with its context and governance intact, to the systems and agents that need it next.

Modernize incrementally

Becoming agent-ready doesn't require rebuilding the data estate from scratch. Instead, this playbook sets out a phased path that builds on the existing platform:

Assess the foundation.

Review existing data sources, lineage, quality tooling, and the current identity model. Identify gaps against the five-layer architecture.

Package reusable data products.

Define ownership, contracts, SLAs, and quality metrics for the highest-value shared assets.

Harden the data foundation.

Embed quality, PII handling, and lineage into ingestion, then establish the lakehouse tier and prepare the first set of unstructured content for retrieval.

Introduce orchestration and the AI gateway.

Establish retrieval services, agent identities, policies, logging, and access controls before putting the first agent into production.

Build the semantic layer.

Define the first ontology with domain owners, resolve entities, and create a knowledge graph for a priority domain.

Close the loop.

Feed agent outputs back into the platform and use operational evidence to improve retrieval quality and outcomes.

Is your data architecture ready for agents?

These five questions will help you identify where the biggest gaps may be:

- → Can you trace an agent response back to the data, policy, and access decision that produced it?
- → Do different agents share the same definition and current state of important business entities?
- → Does unstructured information stay current as its source changes?
- → Can new use cases reuse governed data products rather than building new pipelines?
- → Does every agent have its own identity and least-privilege access?

Build the foundation agents can rely on

The architecture beneath agentic AI has to do more than store and move data. It has to preserve context, establish shared meaning, provide reusable and trustworthy data assets, govern access at the point of use, and make agent activity traceable.

The five layers in this playbook provide a practical reference architecture for doing that without replacing the enterprise data foundation that already exists.

Modernize it layer by layer. Govern it end to end. Give agents the data foundation they need to act with the right context.

Download the full playbook

Talk to us about your data architecture