

On the Design of Robust Industrial Knowledge Graphs: Key Recommendations

Author: Viswanath Avasarala, PhD

Executive Summary

Industrial knowledge graphs have become an essential architectural component for organizations operating complex physical systems. Their primary value lies in providing a shared semantic understanding of assets, systems, and processes across heterogeneous data sources.

In industrial environments, data is distributed across engineering tools, historians, documents, enterprise systems, and analytical platforms. While these systems are individually optimized, they lack a unifying semantic layer that preserves engineering meaning and enables cross-system reasoning. Knowledge graphs address this gap by establishing canonical identity, explicit relationships, and durable semantic structure.

This paper draws on lessons learned from building and operating production-scale knowledge graph systems across multiple industrial organizations. It focuses on architectural choices that consistently determine whether knowledge graphs deliver sustained value or remain fragile, pilot-only implementations. The approach described here also sheds light on the architectural choices underlying DeepIQ's knowledge graph capabilities.

Practitioner Background

My work with ontologies and knowledge representation spans multiple organizations and roles across research, product development, and industrial operations.

Early in my career, I worked as a senior data scientist at GE Research on applied analytics and reasoning systems, collaborating with academic research groups on ontology-driven approaches using large reference models such as the Foundational Model of Anatomy (FMA). This provided early exposure to working with formal semantic models in applied research settings.

I later led the Semantic Web Lab within SAS R&D, where we developed machine-learning-based approaches for rule induction and fact extraction to generate the subject–predicate–object triplets that form knowledge graphs. This work focused on integrating machine learning methods with semantic representations and resulted in patented methods for semantic relationship extraction from natural language (US20160078014A1). Beyond these roles, I have led analytics, data, and semantic initiatives across multiple industrial organizations, with responsibility for designing and

operating production-scale systems supporting analytics, data integration, and AI workflows. Across these environments, I repeatedly encountered the same structural challenges in deploying knowledge graphs at scale. The architectural principles described in this paper emerged from addressing those challenges across different organizations and operational constraints.

1. Why Knowledge Graphs Matter in Industrial Systems

Industrial enterprises operate some of the most complex systems in the world. Assets are long-lived, highly engineered, and embedded within layered physical and organizational structures. Data describing these systems accumulates over decades and is spread across many specialized platforms.

Knowledge graphs provide value in this context by:

- Establishing canonical identity across systems
- Preserving engineering and operational semantics
- Making relationships between assets, systems, and processes explicit
- Providing a stable abstraction layer over heterogeneous data sources

2. Knowledge Graphs and Agentic AI

A major reason industrial knowledge graphs matter right now is the shift from “analytics that report” to “systems that act.” Agentic AI applications are expected to plan, reason, and execute tasks in environments where actions must map back to tangible assets, configurations, and operating context, and where outputs must be defensible.

For these systems, the core technical requirement is **domain depth**: a data layer that grounds an agent or model in domain-specific knowledge with stable identifiers and repeatable semantics.

In industrial settings, three common approaches provide domain grounding:

- **Knowledge graphs**, which represent entities, identity, and relationships explicitly, preserving engineering meaning and enabling traceable context
- **Retrieval-augmented generation (RAG)**, where relevant context is retrieved dynamically from domain data sources at inference time
- **Domain-adapted model training**, where models are trained or fine-tuned directly on curated industrial data

While all three approaches can improve accuracy by reducing ambiguity and increasing access to domain context, knowledge graphs offer distinctive value by providing structured domain context directly to agents or models. This improves grounding and accuracy through canonical entities and explicit relationships, and enables traceability and explanation by making the chain from answer to entity to relationship fully inspectable.

3. Why Many Industrial Knowledge Graph Initiatives Struggle

Despite their conceptual appeal, many industrial knowledge graph initiatives fail to deliver sustained value in production. Common symptoms include:

- High implementation complexity
- Long time to value
- Architectures that are difficult to evolve
- Tight coupling to specific storage technologies

These issues are often attributed to knowledge graphs themselves. In practice, they usually stem from architectural assumptions that are misaligned with the characteristics of industrial data and operational realities.

4. Industrial Knowledge Graphs Are Not Social Graphs

A frequent source of misalignment is the application of design patterns derived from consumer or social-network graphs.

Industrial knowledge graphs differ fundamentally:

- They model engineered systems rather than emergent networks
- Relationships are designed and curated, not inferred from behavior
- Social graphs are optimized for exploiting multi-hop chains of connections, such as mining friend-of-a-friend relationships for recommending new links, whereas industrial knowledge graphs primarily rely on semantically structured paths and generally place less emphasis on deep multi-hop connection expansion.
- The primary goals are traceability, grounding, and decision support

Applying social-network-oriented graph technologies and assumptions to industrial knowledge graphs introduces unnecessary complexity at the storage, query, and analytics layers, without delivering corresponding engineering or operational value.

5. Knowledge Graphs Are Not Graph Databases

Another common misconception is that building a knowledge graph requires adopting a graph database as the primary system of record.

A clear distinction is essential:

- A **knowledge graph** is a semantic construct that defines entities, relationships, meaning, and evolution
- A **graph database** is a storage technology optimized for graph traversal

In many industrial use cases, tightly coupling the knowledge graph to a single graph database undermines the strengths of existing time-series, analytical, and document platforms.

6. Design Choices for Building Knowledge Graphs That Hold Up in Production

Knowledge graphs can be powerful when carefully designed, and deeply problematic when not. In industrial settings, poorly designed knowledge graphs tend to become brittle and costly to evolve. Over time, they accumulate semantic debt, blur the boundary between meaning and data, and become difficult for both humans and systems to reason about.

Based on our experience building and operating large-scale knowledge graph implementations across multiple industrial domains, we arrived at a set of design choices that have consistently held up under production conditions. Many of the perspectives presented below are informed by our work at **DeepIQ**, an industrial DataOps platform that provides comprehensive knowledge graph support for IT–OT contextualization. Working at the intersection of operational technology, enterprise systems, and analytics has repeatedly exposed the practical limits of many commonly adopted knowledge graph design patterns. The principles described below reflect design choices shaped by these constraints, and we elaborate on them here in the hope that they may be useful to other practitioners building and operating industrial knowledge graphs at scale.

6.1 A Semantic Control Plane

In our design, the knowledge graph is treated as a semantic control plane rather than as a primary data store.

The semantic layer models:

- Canonical asset identities
- Asset classes and instances
- Explicit relationships
- Domain schemas and ontologies
- Versioned semantics (for example, slowly changing dimensions)
- Provenance and audit metadata

By keeping this layer independent of how or where raw data is stored, semantics can evolve without forcing disruptive changes to underlying data platforms. In practice, this separation has made systems easier to reason about and maintain.

6.2 Canonical Identity as a First-Class Concern

One of the earliest design decisions we made was to treat canonical identity as foundational.

Industrial data originates from many systems, each with its own identifiers, naming conventions, and lifecycle assumptions. Introducing canonical identities that persist across systems, time, and schema changes has simplified downstream analytics, correlation, and reasoning.

This approach supports:

- Asset-centric analytics
- Cross-system correlation
- Stable grounding for AI models and agents
- Long-term semantic continuity

6.3 A Deliberate Separation from the Data Plane

Enterprises have already invested heavily in data platforms optimized for storage, analytics, and operational workloads. Rather than attempting to replace these systems, our approach allows enterprises to treat existing data platforms as the data plane for the knowledge graph.

Raw data continues to live in platforms designed for specific access patterns, including data lakes, time-series historians, and document or log stores. The knowledge graph provides a semantic layer that references and contextualizes this data without duplicating storage or competing with existing query engines.

By reusing established data platforms, this architecture reduces incremental cost, preserves existing tools and usage patterns, and simplifies change management. Teams adopting similar approaches may find that separating semantic evolution from data platform decisions enables faster adoption and greater long-term flexibility.

6.4 Explicit, Versioned Bindings Between Semantics and Data

To connect semantics to data without tight coupling, we introduced an explicit binding layer between the knowledge graph and the data plane.

These bindings:

- Describe where the data is located
- Define how it can be queried
- Encode index semantics such as time or depth
- Support multiple physical representations over time

Bindings are versioned alongside the semantic model. In practice, resolving graph queries to references in the data plane, rather than returning bulk data directly, has helped keep the semantic layer lightweight and adaptable.

6.5 Versioning and Provenance Built In

From the outset, we treated versioning and provenance as structural properties of the system rather than optional features.

This includes:

- Full semantic version history
- Configuration and schema lineage
- Explicit change attribution (who, when, why)
- Support for safe, non-destructive schema evolution

Embedding these capabilities has been critical for auditability, explainability, and long-lived operation.

6.6 Standards-Aligned and Interoperable Semantics

Where possible, we aligned semantic models with established ontology frameworks and industry standards.

Doing so has made it easier to:

- Incorporate external or reference ontologies
- Maintain consistency with industry standards
- Exchange semantics with other systems
- Preserve the knowledge graph as an enterprise asset rather than a closed implementation

Teams designing similar systems may find that prioritizing interoperability early avoids costly refactoring later.

6.7 Consumption and Access Patterns

The value of a knowledge graph depends on how easily it can be consumed. We focused on simplifying access to semantic context rather than introducing new tools or query paradigms.

The platform supports direct, SQL-based access through the data lake, enabling analysts to work with semantically grounded data using existing workflows. Applications and services access the knowledge graph through a standardized API layer that abstracts persistence while providing a consistent interface to entities and relationships. For

consumers who require direct semantic reasoning, graph-native queries support traversal, and schema-aware exploration.

Agentic AI is treated as a first-class consumer. By exposing an explicit, consistently structured schema layer, the knowledge graph enables agents to interpret domain models generically and to retrieve grounded operational data through standardized access patterns.

In our experience operating production knowledge graph systems, this approach has consistently enabled organic adoption across analytics, AI, and ML workflows, external data sharing, and emerging agentic AI applications.

6.8 Agentic Workflows and Decision Trace Capture

Agentic workflows in the platform can create new knowledge graph instances or update existing ones, with all changes recorded using the exact provenance and versioning mechanisms that govern the rest of the semantic layer. This ensures that agent-driven updates remain attributable, auditable, and consistent with canonical identity and schema evolution.

We are in the process of extending this capability to explicitly model decision trace semantics. Today, agent updates capture what changed and who or what made the change. We are working to fully encode the decision context that led to the outcome, including the evaluated policies, considered alternatives, exceptions, and precedents.

Conclusion

Knowledge graphs can play a valuable role in industrial digital transformation when designed with realistic expectations and aligned with operational constraints.

Most failures arise not from the concept of knowledge graphs, but from architectures that overreach, attempting to replace data platforms, analytical methods, or decision logic. By treating knowledge graphs as a semantic foundation that complements existing techniques such as RAG and enterprise data platforms, organizations can build systems that scale, evolve, and remain trustworthy.
