COMMENTARY

CIMdata®

Embedding AI into PLM-Centric Digital Threads

Technical and architectural best practices for implementing AI across PLM-centric Digital Threads

Takeaways

AI in PLM environments must be grounded in authoritative lifecycle data and governed by digital thread semantics. Retrieval-based approaches tied to product structures, configurations, effectivity, and change context are more reliable than unconstrained generative methods.

Embeddings, vector databases, and retrieval pipelines are lifecycle infrastructure, not implementation details. They require versioning, refresh policies, authorization filtering, and traceability back to source artifacts.

Model orchestration is essential to scale AI responsibly. Different tasks require different models, routing logic, latency targets, and cost controls, which must be explicitly designed and governed.

AI outputs that influence engineering, quality, or compliance decisions must be explainable and auditable by design. This includes visibility into retrieved sources, lifecycle state, configuration scope, and the workflow context in which the output was generated.

Workflow embedding determines whether AI is usable in practice. AI that operates outside change management, requirements, test, quality, and service workflows will not be trusted regardless of technical sophistication.

IP protection and data isolation are architectural constraints. Tenant isolation, role-based access enforcement, controlled persistence, and explicit limits on data reuse must be enforced end-to-end across the AI stack.

Introduction: Technical Challenges Industry Faces

PLM-centric digital threads span a broad range of structured and unstructured data, including product structures, requirements, changes, tests, documents, quality issues, and operational events.[1] These artifacts are typically distributed across multiple systems with inconsistent identifiers, partial integration, and varying lifecycle states. When AI systems are applied without disciplined architectural alignment, these inconsistencies are amplified rather than resolved, undermining trust in AI-generated outputs.

Many early AI implementations extract data from PLM environments into separate AI stacks, introducing duplicate sources of truth, weakened access controls, and broken traceability at the boundary between

[1] Research for this paper was partially supported by CONTACT Software

PLM and AI services. Once lifecycle context is lost, it becomes difficult to explain or defend AI recommendations in engineering, quality, or regulatory settings.

Generative AI tooling has often emphasized conversational interfaces over retrieval discipline, lifecycle awareness, and auditability. While effective in general knowledge domains, these approaches fall short in engineering contexts where configuration scope, effectivity, and approval state determine whether information is valid or actionable.

Organizations frequently lack a defined architecture for managing AI artifacts—models, embeddings, prompts, vector stores, and orchestration logic—as governed lifecycle assets. Without explicit ownership and change control, AI behavior becomes opaque and difficult to scale responsibly.

As a result, AI recommendations are frequently rejected by engineers and quality teams when provenance, rationale, and configuration context cannot be inspected or validated. Data security concerns further compound these challenges, such as tenancy, authorization propagation, logging, caching, and persistence behavior, which directly influence whether AI can be safely deployed in PLM environments.

Some Key AI Term Definitions:

- AI Model: A trained algorithm predicting or generating outputs based on learned patterns.
- Drift: Decline in model performance caused by shifting data or environmental conditions.
- Embeddings: Numerical vectors representing data while preserving its semantic and contextual meaning.
- Extraction: Identifying and pulling specific data points from unstructured text or files.
- Model Context Protocol (MCP): Standardized protocol enabling AI systems to connect with external data sources and tools.
- Orchestration Logic: Control layer coordinating models, tools, and workflows to complete complex tasks.
- Prompts: Structured instructions or queries used to guide and shape model output.
- Retrieval-Augmented Generation (RAG): Using external data to ground model responses in facts.
- Structured Generation: Forcing outputs into specific formats like JSON or predefined schemas.
- Summarization: Condensing content into a shorter version while retaining all essential meanings.
- Vector Store: Database optimized for storing and retrieving embeddings through similarity searches.

Best Practices for Technically Deploying AI Across the Enterprise

Anchor AI to Digital Thread Architecture

AI should be positioned as a service layer within the digital thread rather than as a parallel tool chain. Organizations must define which systems are authoritative for each lifecycle artifact and how AI services are permitted to read, synthesize, and propose outputs.

For each AI use case, the operating boundaries must be explicit, including required lifecycle inputs, authoritative sources, output type, configuration, and effectivity scope, and the governing workflow step. High-impact use cases such as engineering change, compliance, requirements, and test require AI to operate inside governed workflows rather than through free-form entry points.

Architectures that support standardized protocols, such as Model Context Protocol (MCP), can further enable consistent exposure of enterprise data to AI services without introducing brittle, custom integrations.

Retrieval Architecture, Embeddings, and Vector Stores

Retrieval-grounded architecture is essential for PLM use cases. AI responses should be grounded in retrieved, governed artifacts rather than unconstrained model knowledge. Embeddings must be treated as managed pipelines with defined refresh triggers, versioning rules, and retirement of stale vectors. Vector databases should support metadata filtering by lifecycle state, authorization, product, and variant scope, and effectivity to prevent cross-context leakage.

Multi-modal retrieval across documents, requirements, BOMs, tests, and operational records is required, while preserving relationships between lifecycle artifacts.

Orchestration, Prompt Governance, and Observability

Explicit model orchestration is required to scale AI responsibly. Different tasks, such as summarization, extraction, classification, or structured generation, have distinct requirements for determinism, latency, and cost. Prompts and templates must be treated as governed configurations, subject to versioning, testing, and change control when they influence lifecycle decisions. Observability is equally critical; systems should track retrieval hit rates, response quality, latency, cost, and exception patterns to detect drift and failure modes.

Workflow Integration

AI outputs should be embedded as proposed artifacts within existing workflows, such as draft requirements, draft tests, change summaries, compliance evidence packages, or risk analyses. Approval paths must preserve engineering authority, with humans explicitly accepting, rejecting, or revising AI-generated content. Audit records should link AI outputs to retrieved sources, prompts, model versions, and workflow context to support traceability and defensibility.

Closed-Loop Feedback

Downstream signals from manufacturing, service, and quality should be used to improve upstream retrieval and summarization before assuming model fine-tuning or more complex interventions are required. Closed-loop use cases must remain configuration-aware; without a variant and effectivity context, feedback quickly becomes noise. Organizations should institutionalize feedback pipelines rather than relying on ad hoc analysis.

Scale AI Through Measured Execution

AI should be piloted in high-impact areas, validated in practice, and expanded incrementally. Continuous assessment of ROI, adoption, and organizational readiness is required as AI capabilities mature.

IP Protection and Data Security by Architecture

Tenant isolation and least-privilege access must be enforced across all layers, including source systems, retrieval services, vector stores, orchestration, inference, and user interfaces. Persistence rules should explicitly define what is stored, where it is stored, for how long, and for what purpose. CIMdata believes that organizations should avoid AI approaches that require opaque external training or the uncontrolled reuse of customer data outside of governed environments.

CONTACT Software Solution: Fourier AI

Fourier AI's Architecture in CONTACT Elements

Fourier AI is embedded directly within CONTACT Elements, providing AI capabilities inside PLM-centric applications rather than through an external AI platform. Engineers interact with an AI assistant that functions as a knowledgeable colleague with access to product information, design history, and organizational context, augmenting human expertise rather than replacing it.

Fourier AI is architected as a platform capability layer across CONTACT Elements' modular portfolio. AI services sit at the technology layer and are accessible to any module without custom integration, enabling cross-module reasoning across engineering change, requirements, product structure, test management, and manufacturing domains.

Retrieval, Embeddings, and Knowledge Grounding

Fourier AI prepares lifecycle data for AI use by transforming it into embeddings stored in a vector database, enabling semantic search and context-grounded responses. The architecture supports connecting additional enterprise sources, such as intranets or wikis, while raising governance considerations around connector management, access mapping, and refresh policies.

Orchestration, Models, and Prompt Engineering

Fourier AI includes centralized orchestration to dynamically route tasks across AI models based on context, performance, and cost. It orchestrates both commercial large language models and CONTACT's proprietary domain-specific models developed in-house.

An example is CONTACT's multi-modal 3D CAD similarity model, which accepts 2D images, 3D representations, or text queries to identify similar parts across CAD systems using STEP file conversion—capabilities not provided by general-purpose LLMs. Prompt engineering is treated as a core capability, with implied versioning and testing discipline.

Workflow Integration and Evidence of Use

Fourier AI embeds AI interactions directly into the CONTACT Elements user interface, minimizing context switching. In a production deployment at an automotive supplier, Fourier AI generates test cases from PDF requirement documents, creates semantic links, and presents them for engineering approval.

This deployment addresses a bottleneck of approximately 5,000 test cases per year, reducing effort by up to 12 minutes per test case with potential for high five-figure annual savings while standardizing quality. The example demonstrates workflow embedding, governed pipelines, and audit trail integration in a regulated environment.

Security, Tenancy, and Data Handling

Customer data is stored within each customer's own instance and governed by defined roles. The central Fourier Cloud is hosted by CONTACT in AWS under GDPR-compliant European hosting. Information processed by AI models is not stored or reused for other customers or purposes, addressing IP protection and data sovereignty requirements.

The following diagram (Figure 1) provides an Overview of how CONTACT's Fourier AI is architected.

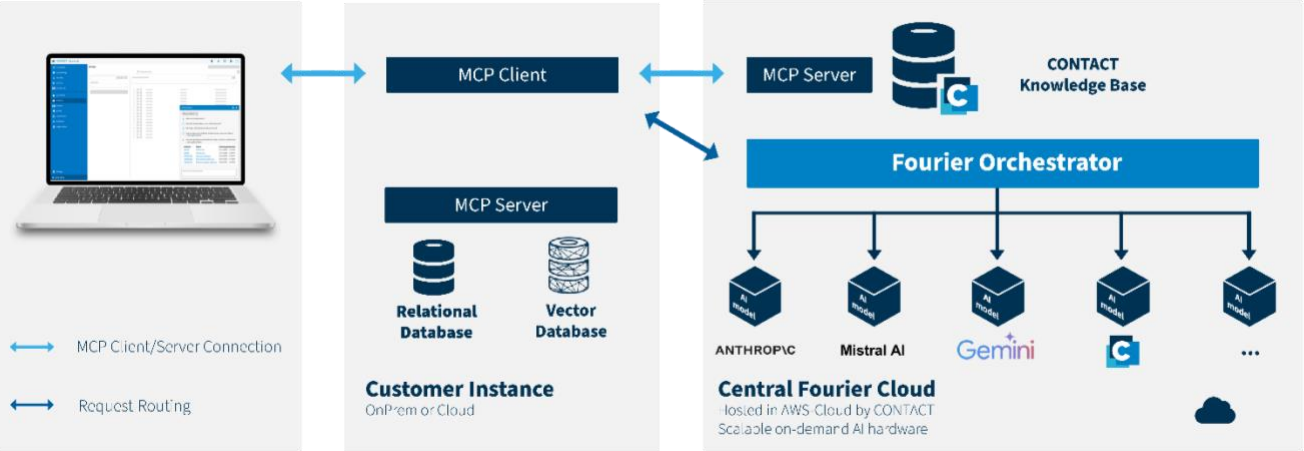

Figure 1: CONTACT Fourier AI Architecture
(Courtesy of CONTACT Software)

Find more information on: CONTACT's Fourier AI webpage

Conclusion

When AI is implemented as a trusted extension of the digital thread, organizations can accelerate decisions while reducing exposure to quality, compliance, and intellectual property risks. Decision makers should view AI as a strategic digital thread capability rather than a standalone technology investment. Successful deployment requires alignment across strategy, PLM-centered data governance, organizational change, and disciplined execution.

For AI to continuously and responsibly scale, it must operate with full lifecycle context, drawing on authoritative product structures, configurations, and change histories while producing outputs that are explainable, traceable, and defensible in engineering and regulatory environments. The greatest long-term value emerges when AI-enabled insights are reinforced through closed-loop feedback from manufacturing, quality, and service, continuously informing upstream decisions. Based on its ability to embed AI into governed lifecycle workflows, support federated architectures, and protect sensitive product data, CIMdata recommends that manufacturers seeking to strengthen their digital threads evaluate CONTACT Software's Fourier AI.

About CIMdata

CIMdata, a global strategic management consulting firm, provides services designed to maximize an enterprise's ability to design, deliver, and support innovative products and services. For more than forty years, CIMdata has provided industrial organizations, providers of digital technologies and services, and investment firms with world-class insight, expertise, and best-practice methods on a broad set of product lifecycle management (PLM) topics and the digital transformation they enable. CIMdata also offers research, subscription services, publications, and education through certificate programs and international conferences. To learn more, visit www.CIMdata.com or email info@CIMdata.com.