WHITE PAPER

CONTEXTUAL ENGINEERING INTELLIGENCE: DATA, APPS, WORKFLOWS WITH AGENTIC AI AND MCP

Infosys®
Navigate your next

FOREWORD

Engineering organizations are navigating a pivotal moment. As product complexity grows and digital transformation accelerates, the ability to manage and leverage engineering data and allied apps/tools effectively has become a strategic differentiator. Yet, many enterprises still struggle with fragmented systems, siloed data, and brittle automation pipelines, which limit agility and innovation.

Agentic AI, a new class of intelligent systems capable of autonomous decision-making, contextual reasoning, and dynamic tool use, offers a breakthrough. These agents can orchestrate complex engineering workflows, adapt to evolving goals, and collaborate across domains with minimal human intervention.

However, autonomy without context is fragile. To unlock the full potential of agentic AI in engineering environments, we need a robust mechanism for sharing and maintaining context across models, tools, and agents. This is where the Model Context Protocol (MCP) comes in. It provides a standardized, interoperable framework for encoding and exchanging contextual information, enabling AI agents to operate with continuity, traceability, and semantic understanding.

This whitepaper presents a perspective on **how the convergence of agentic AI and MCP can transform engineering data and applications.** We explore use cases, contextual patterns, and strategic implications for organizations seeking to lead in the next era of intelligent engineering.

"Unlocking autonomy, context, and collaboration in digital engineering ecosystems."

This statistic underscores the growing reliance on AI in engineering workflows and highlights the need for more **context-aware, interoperable systems** to sustain and scale these gains. Traditional engineering systems are siloed and tool-centric, making it a rigid ecosystem.

1. Fragmentation of Engineering Data and Tools

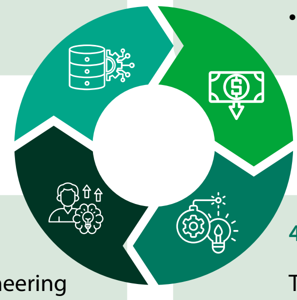
Engineering data is typically distributed across multiple tools and platforms, including Product Lifecycle Management (PLM), Computer-Aided Design (CAD), Computer-Aided Engineering (CAE), and Manufacturing Execution Systems (MES), among others. These systems often lack interoperability, leading to:

- **Data silos** that hinder collaboration across teams and disciplines
- **Manual handoffs** that introduce errors and delays
- **Limited traceability** across the product lifecycle.

2. Rigid Automation and Limited Intelligence

EWhile automation has been introduced in many engineering workflows, it is often rule-based and brittle. Traditional AI/ML models, when applied, are typically narrow in scope and require extensive retraining to adapt to new contexts. This limits their ability to:

- **Generalize** across use cases
- **Adapt to** changing requirements
- **Integrate with** evolving toolchains.



3. The Cost of Context Loss

One of the most significant challenges in engineering systems is the loss of context, i.e., the semantic and operational understanding of how data, models, and decisions relate to one another. Without context:

- **AI systems** struggle to make meaningful inferences
- Engineers spend **significant time** reinterpreting or revalidating data
- **Knowledge** is lost between phases of the product lifecycle.

4. The Need for a New Paradigm

To address these challenges, organizations require an innovative approach that combines intelligent, autonomous agents with a shared, machine-readable context layer.

This is where **Agentic AI** and the Model Context Protocol **(MCP)** come into play, offering a path toward more adaptive, collaborative, and context-aware engineering systems.

CHALLENGES IN INTEROPERABILITY, DATA SILOS AND LEGACY SYSTEMS

Tool Fragmentation | Engineering teams often use a mix of specialized tools like CAD for design, CAE for simulation, PLM for lifecycle management, and MES for manufacturing. These tools:

- Are built by different vendors
- Use proprietary data formats
- Lack of standardized APIs or data exchange protocols.

Data Silos | Engineering data is often stored in isolated repositories. For example,

- Design data in CAD systems
- Simulation results in CAE platforms
- Requirements in document management systems.

These silos prevent holistic analysis and decision-making, requiring manual effort to reconcile and interpret data across domains.

Legacy Systems | Many organizations still rely on legacy systems that:

- Lack cloud-native capabilities
- Are difficult to integrate with modern AI/ML tools
- Require costly maintenance and upgrades.

Legacy infrastructure hinders innovation and makes it harder to adopt intelligent, context-aware solutions.

LIMITATIONS OF TRADITIONAL AUTOMATION IN APPS AND DATA ANALYTICS

Rule-Based Automation | Traditional automation in engineering is often deterministic and rule-based:

- Scripts and macros automate repetitive tasks
- Workflow engines follow predefined paths.

These systems are brittle; they break when inputs change or when exceptions occur, and lack adaptability.

Narrow AI Models | AI/ML models used today are typically:

- Trained on narrow datasets
- Task-specific (e.g., defect detection, predictive maintenance)
- Require retraining for the new context.

They lack the generalization and goal-directed behaviour needed for dynamic engineering environments.

Limited Context Awareness | Most analytics tools operate on isolated datasets without understanding

- The relationships between models and requirements
- The intent behind design decisions
- The lifecycle context of engineering artifacts.

This limits their ability to provide **actionable insights** or support **autonomous decision-making**.

RELEVANCE OF AGENTIC AI IN ENGINEERING DATA, APPS AND WORKFLOWS

As engineering systems become more complex, traditional AI approaches, while helpful, often fall short in dynamic, multi-domain environments. This is where **Agentic AI emerges as a transformative paradigm.** It refers to AI systems that behave as *autonomous agents* capable of perceiving their environment, making decisions, and taking actions to achieve specific goals. Unlike static models, agentic systems are designed to operate continuously, adaptively, and in a contextually relevant manner.

Agentic AI can significantly enhance **various stages of the engineering lifecycle:**

Design:

Agents can assist in generating design alternatives, validating constraints and optimizing parameters.

Simulation:

They can orchestrate simulation runs, interpret results and recommend refinements.

Testing:

Agents can automate test planning, coverage analysis and anomaly detection.

Maintenance:

In operational settings, agents can monitor system health, predict failures and suggest corrective actions.

Agentic **AI differs from Traditional AI/ML** based on tabulated feature extensibilities.

Feature	Traditional AI/ML	Agentic AI
Scope	Narrow, task-specific	Broad, multi-step, goal-oriented
Execu tion	One-shot inference	Continuous, iterative reasoning
Adaptability	Requires retraining	Adapts dynamically using memory and feedback
Tool Interaction	Limited or none	Actively uses tools and APIs
Context Handling	Stateless	Maintains and evolves context over time

In general, traditional AI models are typically trained to perform a single task (e.g., classify images, predict failure). In contrast, agentic AI systems can **plan, reason**, and **act** across a sequence of tasks, making them ideal for complex engineering environments.

Agentic AI is uniquely positioned to address the challenges and unlock new capabilities across the **engineering value chain**. Its ability to reason, act autonomously, and maintain context makes it **a powerful enabler for modern engineering systems.**

Engineering Workflows | Agentic AI can orchestrate and optimize complex, multi-step engineering processes:

Design Automation:
Generate and evaluate design alternatives based on constraints and goals.

Simulation Management:
Automatically configure, run, and interpret simulations across domains.

Test Planning:
Identify gaps in test coverage, prioritize test cases, and adapt plans dynamically.

Change Impact Analysis:
Trace the ripple effects of design or requirement changes across systems.

Engineering Data | Agentic AI can navigate and synthesize diverse data sources:

Contextual Understanding:
Maintain awareness of how data relates to models, requirements and lifecycle stages.

Semantic Linking:
Connect data across silos using ontologies and metadata.

Knowledge Retention:
Build memory over time to avoid redundant work and support continuous learning.

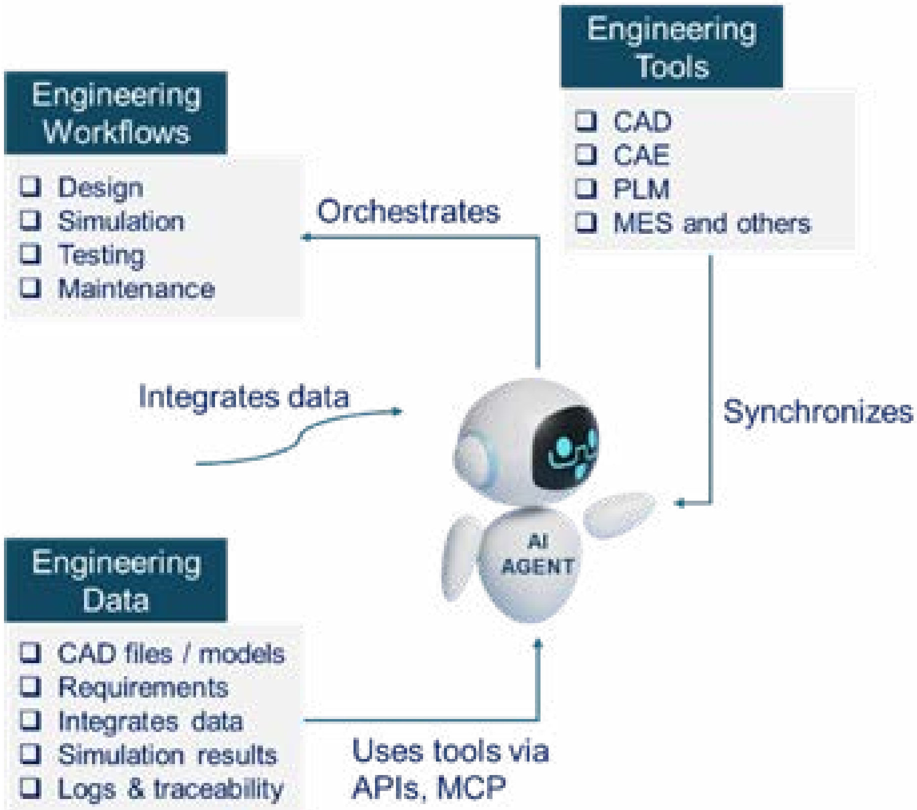
Engineering Tools | Agentic AI can act as a bridge across heterogeneous toolchains:

Tool Orchestration:
Invoke and coordinate actions across CAD, CAE, PLM and MES systems.

API Integration:
Use tool APIs to automate tasks and extract insights.

Cross-Tool Reasoning:
Combine outputs from multiple tools to support higher-level decisions.

By integrating with engineering tools and leveraging protocols like **MCP**, **agentic AI can act as a digital collaborator,** augmenting human engineers, accelerating workflows, and improving decision quality.

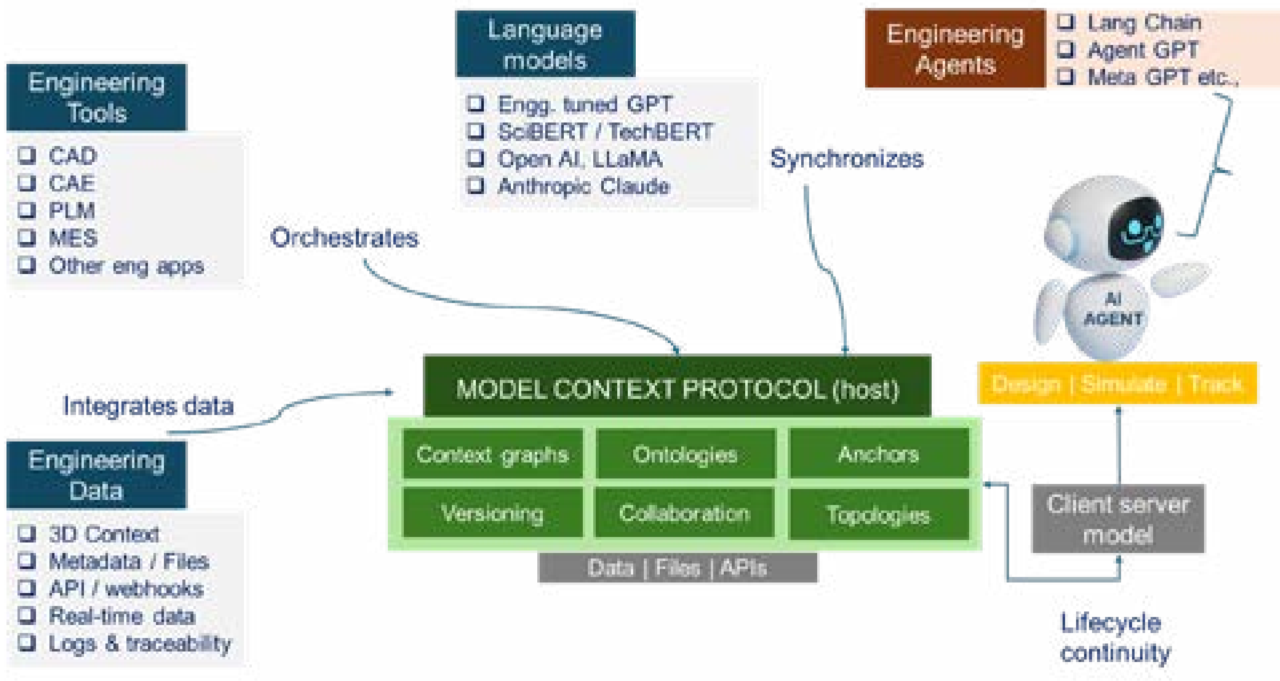


Agentic AI as a Central Orchestrator

ENGINEERING CONTEXTS REIMAGINED WITH AGENTIC AI AND MCP

As engineering systems become more intelligent and interconnected, the need for shared context across tools, models, and AI agents becomes critical. The Model Context Protocol (MCP) addresses this need by **providing a standardized way to represent, exchange, and maintain context throughout the engineering lifecycle.**

The Model Context Protocol **(MCP)** is a machine-readable, interoperable framework designed to encode and share contextual information between systems, models and intelligent agents. It acts as a semantic layer that sits above engineering data and tools, enabling consistent interpretation and traceability.



With MCP acting as a semantic layer, Agentic AI does the orchestration

Key components of MCP include:

Context Graphs:

Structured representations of relationships between models, data, tools and decisions.

Context Anchors:

References that link context to specific artifacts (e.g., a CAD model, a simulation run, a requirement).

Ontologies and Schemas:

Domain-specific vocabularies that ensure semantic consistency.

Versioning and Provenance:

Built-in mechanisms to track changes and maintain historical context.

MCP can be implemented using **graph-based data** structures (e.g., RDF, JSON-LD) and accessed via APIs, making it compatible with modern data platforms and AI systems.

Enabling context sharing across Models and Agents | In traditional systems, context is often implicitly locked inside documents, tool-specific metadata, or tribal knowledge. MCP makes context explicit and portable, enabling:

Cross-Model Understanding:

AI agents can reason across design, simulation, and test models using shared semantics.

Lifecycle Continuity:

Context persists across phases (e.g., from design to manufacturing), reducing rework and misalignment.

Agent Collaboration:

Multiple agents can coordinate tasks by referencing a common context graph.

Explainability and Traceability:

Decisions made by agents or humans can be traced back to the context in which they were made.

DISPLAYING AGENTIC AI WITH MCP IN ENGINEERING KEY DOMAINS

The convergence of Agentic AI and the Model Context Protocol (MCP) opens a wide range of possibilities for transforming engineering workflows. The four illustrative use cases shown below demonstrate **the potential of this paradigm.**

Illustrative Use Case	Challenge	Solution	Impact
Autonomous Design Optimization	Design engineers often manually iterate through CAD models, simulation tools, and requirement documents to optimize a product, an error-prone and time-consuming process.	An agentic AI system, equipped with MCP, can: • Understand design goals and constraints from contextual data • Generate and evaluate multiple design variants using CAD and CAE tools • Maintain traceability of decisions and assumptions via MCP • Recommend optimal configurations based on performance, cost, and manufacturability.	Reduces design cycle time, improves innovation throughput, and ensures alignment with evolving requirements.
Cross-Domain Simulation Orchestration	Running simulations across mechanical, electrical, and thermal domains requires coordination between tools and teams, often with inconsistent data formats and assumptions.	Agentic AI agents can: • Interpret simulation objectives and dependencies using shared context from MCP • Automatically configure and sequence simulations across CAE tools. • Aggregate and analyze results to identify cross-domain interactions • Update the context graph to reflect new insights and dependencies.	Enables faster and more accurate multi-physics simulations, reducing integration errors.
Intelligent Maintenance and Diagnostics	Maintenance teams struggle to interpret sensor data, logs, and historical records to predict failures or recommend actions.	An agentic AI system can: • Continuously monitor operational data streams • Use MCP to correlate real-time data with historical context (e.g., design intent, past failures) • Predict potential issues and suggest corrective actions • Trigger automated workflows for inspection or part replacement.	Improves asset uptime, reduces unplanned downtime, and enhances safety.
Engineering Knowledge Management and Reuse	Valuable engineering knowledge is often lost in documents, emails, or retired systems, making it hard to reuse past work.	Agentic AI agents can: • Extract and structure knowledge from legacy documents and systems • Use MCP to link knowledge to relevant models, tools, and lifecycle stages • Answer queries, suggest design patterns, or flag known issues during new projects.	Accelerates onboarding, reduces duplication of effort, and preserves institutional knowledge.

BENEFITS AND IMPACT

The integration of Agentic AI and the Model Context Protocol (MCP) represents a paradigm shift in how **engineering organizations manage complexity, drive innovation, and scale intelligence** across the product lifecycle. Below are the key benefits and their impact across technical and business dimensions.

- **Faster design iterations through** autonomous optimization and simulation orchestration
- **Reduced manual effort** in data preparation, validation, and tool integration. **Improved traceability** of decisions across the lifecycle|
- **Shorter time-to-market** for complex products
- **Context-aware insights** from AI agents that understand the relationships between models, data and requirements
- **Real-time recommendations** based on historical knowledge, operational data and simulation results
- **Cross-tool collaboration** enabled by MCP's semantic context layer

- **Reduced integration overhead** between legacy and modern systems
- **Standardized data exchange** across domains and disciplines
- **Multi-agent orchestration** of complex workflows
- **Dynamic adaptation** to changing goals, **Reusable automation patterns** across projects and teams
- **Institutional memory** captured and structured via MCP with reduced redundancies and efforts
- **Faster onboarding** of new engineers through intelligent knowledge access
- **Future-proof architecture** that supports evolving tools, models and AI capabilities

CONCLUSION

Engineering organizations are at a crossroads. The increasing complexity of products, data, and tools demands a new approach, one that goes beyond traditional automation and analytics. **Agentic AI,** empowered by the **Model Context Protocol (MCP)**, offers a transformative solution: intelligent systems that can reason, act, and collaborate across the engineering lifecycle with shared context and autonomy.

This convergence enables:

- Smarter, faster decision-making
- Seamless integration across tools and domains
- Scalable automation and knowledge reuse.

Together, Agentic AI and MCP redefine what's possible in digital engineering, establishing contextual intelligence across data, apps and workflows.

FOREWORD AND CONCLUSION BY

Nandish Ramanna

Senior Project Manager, Infosys Limited

About the Authors

Baskar Ramamurthy

Project Manager, Infosys Limited

Raghavendar Kannan

Engineering Lead, Infosys Limited

For more information, contact askus@infosys.com

Infosys®
Navigate your next

Infosys.com | NYSE: INFY

Stay Connected