Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Context Engineering for Enterprise AI: Architecting Persistent, Governed Intelligence in Regulated Industries

Shikhar Singhal

Boston University, USA

ARTICLE INFO

Received: 28 Jan 2026

Accepted: 02 Feb 2026

ABSTRACT

Large language models demonstrate powerful reasoning capabilities across diverse domains. Enterprise deployments face persistent challenges when scaling beyond isolated use cases. Most implementations fail due to inadequate context management and governance frameworks. Traditional prompt-centric designs treat models as stateless reasoning engines without persistent memory. Each interaction starts fresh without access to historical decisions or institutional knowledge. Regulated industries require explainability, auditability, and jurisdiction-aware compliance. Current architectures lack systematic mechanisms to enforce these requirements. This article introduces Context Engineering as a foundational architectural discipline for enterprise artificial intelligence systems. Context becomes a first-class system component rather than prompt-level input data. The framework integrates structured and unstructured enterprise data through controlled memory stores and permissioned pipelines. Governance mechanisms are embedded directly into context construction and validation processes. Compliance rules operate proactively during context assembly rather than through reactive validation. Enterprise insurance implementations demonstrate improved decision consistency and reduced hallucination rates. The architecture enables scalable human-artificial intelligence collaboration while maintaining regulatory compliance. Intelligence emerges as an engineered system property grounded in persistent context rather than solely as model capability.

Keywords: Context Engineering, Enterprise AI Architecture, Governance Frameworks, Regulated Industries, Persistent Intelligence, Compliance by Design, Human-AI Collaboration

1. Introduction

Enterprise adoption of large language models accelerates across financial services, healthcare, and insurance sectors. Organizations seek to leverage generative artificial intelligence for decision support and process automation. Most deployments encounter significant barriers when transitioning to production systems. The fundamental challenge lies in architectural design patterns rather than model performance limitations.

Traditional implementations treat models as stateless reasoning engines. Each interaction begins with minimal context through prompt engineering techniques. This approach works for simple tasks but

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

breaks down in complex enterprise environments. Business processes require access to historical decisions and organizational knowledge. Chain-of-thought prompting has demonstrated that language models can perform complex reasoning when provided with intermediate steps [1]. However, prompt-based designs cannot effectively embed a comprehensive institutional context within token limits.

Regulated industries face additional constraints beyond technical performance requirements. The fundamental understanding of the above statement is that companies in the financial services sector must meet a variety of governance frameworks, including the Dodd-Frank Act, while also meeting state-level regulations for insurance companies and complying with HIPAA regulations for healthcare organizations. Every artificial intelligence-generated decision requires explainability and auditability. Current architectures lack systematic mechanisms to enforce compliance requirements. The evolution of ChatGPT and related large language models has highlighted both opportunities and significant limitations in enterprise contexts [2].

This article proposes Context Engineering as a distinct architectural discipline for enterprise artificial intelligence systems. Context elevates from prompt-level concern to system-level component with explicit design and governance. The framework formalizes context as a structured representation encompassing operational state, business rules, regulatory constraints, and decision history. Persistent intelligence emerges through controlled integration of enterprise data sources and memory stores.

The framework addresses three critical gaps in current enterprise deployments. Persistent context maintains state across interactions while preserving institutional knowledge. Embedded governance enforces regulatory requirements during context construction rather than through external validation. Permissioned access patterns respect data boundaries and jurisdiction-specific regulations through architectural controls. These capabilities enable scalable deployment in regulated environments.

The article is organized into five additional sections. Section 2 establishes Context Engineering as an architectural discipline distinct from prompt engineering. Section 3 presents the framework architecture and core components. Section 4 details governance mechanisms and compliance integration. Section 5 demonstrates the approach through enterprise insurance use cases. Section 6 concludes with implications for enterprise artificial intelligence architecture.

2. Context Engineering as an Architectural Discipline

2.1 Limitations of Prompt-Centric Approaches

Most enterprise artificial intelligence implementations rely heavily on prompt engineering to guide model behavior. Context is provided through carefully crafted instructions embedded in each request. While prompt engineering delivers value for isolated tasks, it introduces fundamental limitations for enterprise systems.

Prompt-centric designs are inherently stateless without memory of prior decisions. Business processes spanning multiple steps lose continuity between interactions. A claims adjuster reviewing complex insurance cases cannot maintain context across multiple artificial intelligence consultations. The system rebuilds understanding from scratch with each query. This pattern leads to inconsistent recommendations and reduced efficiency.

Token limits create additional constraints on prompt-based context assembly. Enterprise decisions often require extensive background information, including policy documents and regulatory guidelines. Attempting to embed all relevant context into prompts quickly exceeds practical token budgets. Organizations face impossible choices between completeness and feasibility when designing prompts.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Federated learning frameworks have shown promise for privacy-preserving machine learning across distributed data sources [3]. However, these approaches do not address the fundamental challenge of context persistence in enterprise artificial intelligence systems. Traditional prompt engineering cannot leverage federated architectures effectively without systematic context management.

Governance and compliance requirements cannot be effectively managed through prompts alone. Regulatory constraints must be enforced consistently across all interactions. Embedding compliance logic in prompts creates maintenance challenges and inconsistency risks. Changes to regulations require updates across potentially hundreds of prompt templates. The approach does not scale for organizations operating across multiple jurisdictions with varying requirements.

Prompt engineering struggles with knowledge evolution and organizational learning. Business rules change over time as regulations evolve. Organizations accumulate experience through historical decisions. ChatGPT and similar models have demonstrated both significant capabilities and notable limitations, including hallucinations and inconsistent outputs [4]. Prompt-based systems lack mechanisms to systematically incorporate new knowledge while maintaining historical context. Updates require manual prompt revisions rather than systematic knowledge integration.

2.2 Context as a First-Class System Component

Context Engineering reframes context as an explicit architectural component with dedicated design and management. Rather than treating context as input data embedded in prompts, the framework establishes context as a persistent system entity. This elevation transforms how enterprise artificial intelligence systems are architected and operated.

The framework defines context through multiple layers of structured representation. Operational context captures the current system state and active workflows. Business context encodes organizational knowledge, including policies and domain expertise. Regulatory context maintains jurisdiction-specific constraints and compliance requirements. Historical context preserves decision trails and accumulated institutional learning. Each layer serves distinct purposes while integrating into a coherent system context.

Persistent context enables continuity across interactions and workflows. The system maintains state between artificial intelligence consultations while accumulating knowledge. A multi-day underwriting process retains context as different specialists contribute their expertise. The artificial intelligence system builds understanding progressively rather than starting fresh with each query. This persistence fundamentally changes how artificial intelligence integrates into business processes.

Structured context representation enables systematic governance and control mechanisms. Compliance rules are encoded as explicit constraints on context construction. Regulatory requirements are enforced through architectural mechanisms rather than prompt instructions. The system verifies that all necessary governance checks have been applied before generating outputs. This capability provides auditability and explainability that prompt-based designs cannot achieve.

Context Engineering separates concerns that prompt-centric approaches conflate. Model reasoning capabilities are distinct from context management responsibilities. The language model performs inference based on the provided context. Context construction and governance are handled by dedicated system components. This separation enables independent evolution of reasoning engines and enterprise integration logic.

2.3 Architectural Requirements for Regulated Environments

Enterprise artificial intelligence systems in regulated industries must satisfy requirements beyond model performance metrics. These constraints shape architectural decisions and drive the need for

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

engineered context. Financial services firms operate under complex regulatory frameworks. Insurance companies face state-specific regulations. Healthcare organizations comply with strict privacy requirements.

Explainability requirements demand that every artificial intelligence-generated recommendation can be traced to specific evidence. Regulators and auditors must reconstruct how decisions were reached. Prompt-based systems provide limited visibility into context influencing outputs. Engineered context enables comprehensive audit trails by maintaining structured records of all information considered during inference.

Compliance mechanisms must be embedded systematically rather than enforced through external validation. Waiting until after inference to check regulatory compliance introduces the risk of policy violations. Context Engineering enforces constraints during the context construction phase. Only compliant information reaches the reasoning engine. Jurisdiction-aware context pipelines automatically apply appropriate regulatory filters based on transaction geography.

The Governance Requirements Related to Data Governance Define What Information Can Be Combined to Achieve a Specific Business Outcome. The Privacy Regulations Define the Cross-Border Data Flow Limitations. The Confidentiality Regulations Define the Strategy for Separate Businesses Being Operated/Managed under Confidentiality.

The Above Regulations Must Be Considered When Organizing Contexts. The Data Governance Related to Defined Contexts Provides For Only the Combination of Pre-approved Data for AI Reasoning-AI Will Only Operate With Authorized Data.

There Is a Necessity for Human Oversight in the Cases Where AI Will Be Supporting High Stakes Decisions in Regulated Industries. To Keep the AI in the Context of the Human Judgment Is a Critical Requirement. Context Engineering facilitates human-artificial intelligence collaboration by providing shared context. Decision support interfaces present the same context used for artificial intelligence inference. Reviewers can validate recommendations against source evidence.

ARCHITECTURAL DIMENSION	PROMPT-CENTRIC APPROACH	CONTEXT-ENGINEERED APPROACH
State Management	Stateless interactions with no memory persistence	Persistent context maintained across interactions and workflows
Governance Model	Compliance embedded in prompt templates requiring manual updates	Systematic enforcement through dedicated architectural components
Knowledge Evolution	Manual prompt revisions for each organizational change	Versioned context store with automated propagation of updates
Context Scope	Limited by token constraints and single-prompt embedding	Multi-tiered memory architecture with comprehensive knowledge integration
Audit Capability	Limited visibility into reasoning process and context sources	Complete provenance tracking with structured audit trails

Table 1: Comparison of Prompt-Centric and Context-Engineered Approaches [3, 4]

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3. Framework Architecture and Components

3.1 Context Store and Memory Architecture

The Context Store serves as the foundational component for persistent intelligence in enterprise artificial intelligence systems. This specialized data architecture maintains structured representations of operational state, business knowledge, and decision history. The Context Store is designed specifically for context retrieval and composition during artificial intelligence inference.

The architecture employs multi-tiered memory with different persistence characteristics. Short-term context maintains active workflow state and recent interaction history. This layer provides immediate context for ongoing processes. Medium-term context preserves completed workflows and accumulated knowledge from recent operations. Long Term Context Is A Repository for Business Policies And Governance Frameworks That Span A Long Term Timeframe.

In Research Conducted By Schools of Higher Learning Related to Large Language Models, Institutions Have Identified Significant High-Risk Opportunities, As Well As High-Risk Challenges, Related to The Areas of Bias, Privacy, And Ethical Use Of Information. In Enterprise Contexts, Institutional Memory And Governance Will Have An Increased Impact. The Context Store addresses these challenges through systematic architecture rather than ad-hoc solutions.

Context representation combines structured and unstructured data formats. Structured context includes entity relationships and business rules that enable precise retrieval. Unstructured context encompasses policy documents and historical case notes that provide a nuanced understanding. The hybrid format allows the system to leverage both precise logical reasoning and contextual understanding.

Versioning mechanisms track context evolution over time. Business rules change as regulations are updated. Organizational knowledge expands through experience. The Context Store maintains version history to support temporal reasoning and audit requirements. Systems can reconstruct historical context to explain past decisions.

Retrieval mechanisms optimize for relevance and completeness during context assembly. Semantic search identifies related contexts based on conceptual similarity. Graph traversal follows relationship chains to gather connected context elements. Temporal filtering selects a context appropriate for specific time periods. These capabilities enable the construction of a comprehensive yet focused context.

3.2 Context Pipeline and Integration Layer

The Context Pipeline orchestrates how enterprise data flows into a structured context representation. This integration layer connects distributed data sources while enforcing governance rules. The pipeline architecture ensures that context assembly respects data boundaries and maintains audit trails.

Source adapters provide standardized interfaces to heterogeneous enterprise systems. These components extract relevant information from policy administration systems and claims databases. Each adapter understands source-specific data models. Information is exposed through common integration interfaces.

Large language models in medicine demonstrate both transformative potential and significant risks that require careful governance [6]. The same principles apply to enterprise contexts where clinical decision support, regulatory compliance, and patient privacy intersect. Context pipelines must enforce domain-specific requirements through architectural controls.

Transformation components enrich and normalize data from multiple sources into a consistent context representation. Entity resolution identifies when records from different systems refer to the same

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

customer or policy. Data validation applies business rules to ensure quality and completeness. Semantic annotation adds metadata that supports later reasoning and retrieval.

Governance gates embedded in the pipeline enforce regulatory and business constraints during context assembly. Data classification determines sensitivity levels and applicable restrictions. Access control verifies that the requested context is authorized for specific users. Compliance filters remove information that cannot be used under applicable regulations. These gates operate automatically based on configurable policy rules.

Context composition assembles the final context from processed data based on specific inference requirements. The composition layer selects relevant information and organizes it according to defined schemas. This final step produces complete context packages ready for artificial intelligence reasoning. Intelligent selection and summarization keep context within token limits.

3.3 Permissioned Context Access

Enterprise artificial intelligence systems must respect complex access control requirements governing information use. Artificial Intelligence (AI) is only able to reason with data that has been approved for use by a person or entity due to the enforcement of context access; therefore, it is critical to have this function, so it does not violate privacy regulations or organizational policies regarding data governance.

Role-based access control defines what context each user can access based on organizational position. A claims adjuster receives context relevant to assigned cases. Underwriters see different context subsets than actuaries despite working with related information. These boundaries are enforced during context retrieval and composition.

Purpose-based restrictions limit context use to authorized applications. Information collected for one business purpose cannot automatically be used for other purposes. Customer data gathered for claims processing may not be available for marketing analytics. The context system tracks data provenance and enforces purpose restrictions.

Jurisdiction-aware access control applies geographic restrictions on data access. Privacy regulations impose territorial boundaries on information use. Context pipelines automatically filter out data that cannot cross applicable jurisdictional boundaries. This capability enables organizations to operate globally while maintaining compliance.

Temporal access control limits the availability of certain contexts based on time constraints. Historical decisions may become sealed after specific retention periods. Preliminary results might be restricted until the formal review is completed. Time-based access rules ensure that context availability aligns with business requirements.

Dynamic permission evaluation adjusts access based on runtime conditions rather than static rules alone. Emergency access protocols temporarily expand context availability during urgent situations. Escalation workflows modify permissions as cases move through approval chains. This flexibility enables practical operations while maintaining governance principles.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

COMPONENT LAYER	PRIMARY FUNCTION	OPERATIONAL CHARACTERISTICS
Short-Term Context	Maintains active workflow state and recent interaction history	Immediate availability for ongoing processes with rapid access patterns
Medium-Term Context	Preserves completed workflows and accumulated operational knowledge	Retention of recent business operations supporting pattern recognition
Long-Term Context	Stores organizational policies and regulatory frameworks	Extended persistence of institutional knowledge and compliance requirements
Retrieval Mechanisms	Semantic search and graph traversal for context assembly	Relevance optimization through conceptual similarity and relationship chains
Versioning System	Tracks context evolution and maintains historical snapshots	Temporal accuracy for audit purposes and regulatory reconstruction

Table 2: Context Store Architecture Components and Functions [5, 6]

4. Governance and Compliance Mechanisms

4.1 Embedded Compliance Architecture

Traditional approaches treat compliance as external validation applied after artificial intelligence inference. This reactive model introduces the risk of policy violations. Context Engineering embeds compliance directly into system architecture through proactive mechanisms. Non-compliant contexts are prevented from reaching reasoning engines.

Compliance rules are encoded as executable policies that operate during context construction. These policies express regulatory requirements and organizational guidelines in forms that systems can enforce automatically. Financial services firms encode policies preventing use of certain customer attributes in credit decisions. Insurance companies enforce restrictions on discriminatory factors in underwriting recommendations.

Policy evaluation occurs at multiple stages throughout context pipelines. Source adapters apply initial filtering based on data classification. Transformation components verify that enrichment operations comply with data combination restrictions. Context composition performs final validation before delivering context for inference. The Defense-in-Depth method of artificial intelligence provides a comprehensive approach to ensure that there are no gaps in coverage.

Upon examining the categories of failure types from ChatGPT, can identify similar failure modes, such as poor reasoning or acronym errors, and inappropriate content generation. Ultimately, failure modes indicate that governance should be specified in the design of the AI system as opposed to relying solely on the AI model for governance. Context Engineering addresses these vulnerabilities through architectural controls.

Jurisdiction detection automatically determines which regulatory frameworks apply to each context assembly operation. Geographic indicators from customer addresses and transaction locations trigger

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

appropriate compliance policies. The system seamlessly applies different requirements for European customers versus other jurisdictions. This automatic adaptation enables global operations without manual policy management.

Compliance evidence is captured automatically during policy evaluation. The system records which rules were evaluated and what checks were performed. These records support regulatory examinations and internal audits. Evidence is linked directly to specific inference operations. Complete compliance verification can be reconstructed for any artificial intelligence-generated recommendation.

4.2 Explainability and Audit Trails

Regulatory examinations require comprehensive records of how artificial intelligence systems reached specific decisions. Context Engineering provides explainability through detailed tracking of context assembly and inference inputs. These capabilities transform opaque artificial intelligence operations into transparent decision workflows.

Context provenance tracking records the origin of every piece of information included in inference context. The system maintains lineage from original source systems through all transformation steps. An auditor examining an underwriting recommendation can trace risk factors back to specific policy applications. This transparency enables verification that only appropriate information influenced decisions.

Financial services face unique artificial intelligence challenges, including model risk management, data quality, and regulatory compliance requirements [8]. These constraints demand systematic audit capabilities beyond what traditional artificial intelligence architectures provide. Context Engineering addresses financial sector requirements through comprehensive provenance tracking.

Inference logging captures the complete context provided to language models along with generated outputs. These records preserve exact inputs and results for later review. The system can reconstruct the precise information state that existed during any historical inference operation. Temporal consistency is maintained even as underlying policies change.

Decision justification mechanisms annotate artificial intelligence-generated recommendations with explicit references to supporting evidence. Rather than producing standalone conclusions, the system links each claim to specific context elements. An approval recommendation cites relevant policy provisions and precedent cases. This citation enables human reviewers to validate reasoning.

Structured audit logs record all context access and policy evaluations. These logs use standardized formats that support automated reporting. Pattern detection identifies anomalous context usage or policy violations. Aggregate reporting demonstrates compliance across thousands of inference operations.

4.3 Bias Detection and Fairness Controls

Enterprise artificial intelligence systems in regulated industries must detect and mitigate unfair bias in decision recommendations. Bias can emerge from training data or context selection. Context Engineering addresses bias through architectural controls that operate during context assembly rather than relying solely on post-hoc validation.

Context diversity assessment evaluates whether information included in the inference context provides a balanced representation. The system detects when context is dominated by examples from specific demographic groups. Alerts trigger when diversity metrics fall below configured thresholds. This proactive approach prevents biased context from influencing recommendations.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Sensitive attribute filtering removes protected characteristics from context when their use would violate anti-discrimination regulations. The system identifies attributes related to race, gender, and age. These attributes are excluded from context unless explicitly authorized for legitimate business purposes with documented justification. Filtering occurs automatically based on jurisdiction-specific policies.

Counterfactual testing evaluates how decisions change when sensitive attributes are modified. The system generates synthetic context variations that alter demographic characteristics while keeping other factors constant. Recommendations should remain stable across these variations. Significant changes indicate potential bias requiring investigation.

Fairness metrics are computed across cohorts to identify disparate impact. The system analyzes historical recommendations to detect patterns where specific groups receive systematically different treatment. Statistical testing determines whether observed disparities exceed thresholds consistent with chance. Detected issues trigger a review of context construction rules.

Remediation workflows provide structured processes for addressing detected bias. When fairness concerns are identified, designated reviewers examine context selection rules and data quality. Corrections are implemented systematically through pipeline configuration rather than ad-hoc adjustments. All remediation actions are documented for audit purposes.

CONTROL POINT	GOVERNANCE FUNCTION	ENFORCEMENT MECHANISM
Source Adapters	Initial data classification and sensitivity assessment	Automatic filtering based on regulatory data categories and access permissions
Transformation Layer	Validation of data combination and enrichment operations	Business rule verification ensuring compliance with jurisdiction-specific restrictions
Context Composition	Final validation before inference delivery	Comprehensive policy evaluation confirming all governance checks completed
Jurisdiction Detection	Geographic determination of applicable regulatory frameworks	Automatic triggering of region-specific compliance policies based on transaction indicators
Compliance Evidence	Capture and documentation of policy evaluations	Structured recording of all governance actions linked to specific inference operations

Table 3: Embedded Governance and Compliance Control Points [7, 8]

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5. Enterprise Insurance Use Cases

5.1 Claims Adjudication and Decision Support

Claims adjudication is a complex process that requires integrating the regulatory requirements associated with an insurance policy, the terms and conditions of that policy, and the medical evidence. Traditionally, this process has been manually performed and has been both time-consuming and inconsistent. In addition, there are concerns about the ability of a fully automated process to remain compliant with both regulatory requirements and governance policies. Context-engineered artificial intelligence systems enable a hybrid model that augments human expertise.

The Context Store maintains comprehensive claim context across extended adjudication timelines. Initial loss reports and medical documentation accumulate as cases progress. The system preserves chronological records of decision points and rationale. Each adjuster consultation builds on previous insights rather than starting fresh.

Context assembly for claims decisions integrates multiple specialized knowledge domains. Policy interpretation context includes specific contract language and coverage exclusions. The medical context provides relevant clinical guidelines. Legal context incorporates jurisdiction-specific regulations and precedent decisions. Fraud detection context includes risk indicators and historical patterns.

Comprehensive surveys of large language model applications identify insurance claims processing as a high-potential use case with significant technical and regulatory challenges [9]. Context Engineering specifically addresses these challenges through systematic governance and persistent memory architecture.

Governance mechanisms ensure compliant decision support throughout the adjudication process. The system validates that all required documentation has been collected before recommending payment decisions. Regulatory filters ensure compliance with state-specific claims handling requirements. Approval thresholds trigger additional review for high-value cases. These controls are embedded in context validation.

Decision consistency improves substantially through persistent context and systematic reasoning. Similar claims receive comparable treatment because the system maintains a structured memory of precedent decisions. Adjusters receive consistent guidance even when different individuals handle related claims. This consistency reduces errors and supports fair treatment.

5.2 Underwriting and Risk Assessment

Underwriting decisions require the synthesis of applicant information, actuarial models, and regulatory constraints. Context-engineered systems transform this synthesis from manual data gathering to systematic knowledge integration. The architecture enables consistent risk assessment while respecting complex access controls.

Application context is assembled from multiple source systems, including quote platforms and external data providers. The Context Pipeline normalizes this heterogeneous data into structured representations. Entity resolution links current applications to existing policies. Data validation ensures completeness and flags inconsistencies requiring manual review.

Risk assessment context includes actuarial models and underwriting guidelines. These components are maintained as versioned knowledge in the Context Store. Model updates propagate systematically rather than requiring manual policy revisions. Historical context enables experience-based adjustments while maintaining compliance with regulatory restrictions.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Multitask evaluations of ChatGPT reveal significant challenges with reasoning consistency, factual hallucination, and interactive dialogue management [10]. These limitations are particularly problematic for underwriting, where accuracy and reliability are essential. Context Engineering mitigates these risks through structured context validation and governance controls.

Permissioned access controls respect regulatory boundaries on information use. Certain rating factors are prohibited in specific jurisdictions. Some data sources can only be used with explicit applicant consent. The context system enforces these restrictions during assembly. Jurisdiction-aware pipelines automatically apply appropriate filters[11].

Context persistence enables multi-stage underwriting workflows. Initial automated screening uses basic context to route applications. Detailed underwriting incorporates specialized inspections as results become available. Senior review adds expert judgment to complex cases. The system maintains coherent context across these stages while controlling access[12].

5.3 Regulatory Reporting and Compliance

Insurance businesses are subject to extensive regulatory reporting requirements from a multitude of jurisdictions. Compliance teams must aggregate data and validate accuracy. Context-engineered systems streamline regulatory compliance by maintaining structured audit trails and enabling systematic evidence generation.

Compliance context is assembled continuously rather than reactively when reports are due. The Context Store maintains structured records of all policies and claims. Governance mechanisms capture evidence of regulatory compliance during operational processes. This continuous capture eliminates manual reconstruction of compliance history.

Jurisdiction-specific reporting pipelines automatically configure for applicable regulatory requirements. A multi-state insurer operates different context filters for each state regulator. Federal requirements are layered on top of state-specific rules. The system manages this complexity through composable policy configurations[12].

Before filing a return with the appropriate authority, validation mechanisms verify the completeness and accuracy of the data being submitted. A validation system verifies that all required data elements exist and are correctly formatted. Cross-validation identifies inconsistencies between related reporting fields. Historical comparison flags anomalous changes requiring explanation. These automated checks reduce filing errors[13].

Examination support capabilities enable efficient response to regulatory inquiries. The context system can rapidly retrieve records demonstrating compliance with specific requirements. Audit trails document complete decision processes for sample cases selected by regulators. Evidence packages are assembled automatically from structured compliance records.

Regulatory change management is streamlined through systematic context updates. When new requirements take effect, compliance teams update policy configurations in the Context Store. The system automatically applies updated rules to new operations while maintaining historical context. This versioning ensures temporal accuracy for audit purposes[14]].

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

IMPLEMENTATION DOMAIN	CONTEXT ENGINEERING APPLICATION	GOVERNANCE INTEGRATION
Claims Adjudication	Persistent case context across extended timelines with multi-domain knowledge synthesis	State-specific regulatory compliance and approval threshold enforcement
Underwriting Assessment	Multi-stage workflow support with actuarial model integration and entity resolution	Jurisdiction-aware rating factor restrictions and sensitive attribute filtering
Risk Evaluation	Historical loss experience integration with versioned underwriting guidelines	Bias detection through counterfactual testing and diversity analysis
Regulatory Reporting	Continuous compliance context assembly with automated evidence generation	Multi-jurisdiction pipeline configuration and temporal accuracy maintenance
Fraud Detection	Pattern recognition across historical claims with risk indicator tracking	Privacy-preserving analysis within permissioned access boundaries

Table 4: Enterprise Insurance Use Case Implementation Domains [9, 10]

Conclusion

Enterprise artificial intelligence deployment in regulated industries requires architectural frameworks that extend beyond model capabilities to encompass systematic context management and embedded governance. Context Engineering addresses fundamental limitations of prompt-centric and stateless artificial intelligence designs. The framework establishes context as a first-class system component with explicit design and compliance controls. Persistent context maintains operational state and business knowledge across interactions. This continuity enables progressive understanding that stateless designs cannot achieve. Embedded governance mechanisms enforce regulatory requirements during context construction rather than treating compliance as external validation. This proactive model reduces risk and streamlines audit processes. Permissioned access patterns respect data boundaries through architectural controls that operate automatically based on jurisdiction and role. Enterprise insurance implementations demonstrate practical value for claims adjudication, underwriting, and regulatory compliance. Organizations achieve improved decision consistency and enhanced regulatory compliance compared to manual processes. The architecture scales across diverse business processes while maintaining governance principles. Context-engineered systems offer a path toward reliable and compliant enterprise intelligence as generative artificial intelligence adoption accelerates. Thus, intelligence as an engineered characteristic of a system is realized through the preservation of context, rather than through the capacity of the model alone. Organizations that adopt these architectural

principles will be better positioned to deploy artificial intelligence systems that deliver sustained value while meeting regulatory requirements.

References

[1] Jason Wei, et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," arXiv, 2023. Available: https://arxiv.org/abs/2201.11903

[2] Yiheng Liu, et al., "Summary of ChatGPT-Related research and perspective towards the future of large language models," Meta-Radiology, 2023. Available: https://www.sciencedirect.com/science/article/pii/S2950162823000176

[3] Qiang Yang, "Toward Responsible AI: An Overview of Federated Learning for User-centered Privacy-preserving Computing," ACM Digital Library, 2021. Available:https://dl.acm.org/doi/abs/10.1145/3485875?sid=SCITRUS

[4] Aram Bahrini, et al., "ChatGPT: Applications, Opportunities, and Threats," arXiv, 2023. Available: https://arxiv.org/abs/2304.09103

[5] Enkelejda Kasneci, et al., "ChatGPT for good? On opportunities and challenges of large language models for education," Learning and Individual Differences, 2023. Available: https://www.sciencedirect.com/science/article/abs/pii/S1041608023000195

[6] Arun James Thirunavukarasu, et al., "Large language models in medicine," Nature Medicine, 2023. Available: https://pubmed.ncbi.nlm.nih.gov/37460753/

[7] Ali Borji, "A Categorical Archive of ChatGPT Failures," arXiv, 2023.Available: https://arxiv.org/pdf/2302.03494

[8] Financial Stability Board, "Artificial intelligence and machine learning in financial services: Market developments and financial stability implications," 2017. Available: https://www.fsb.org/uploads/P011117.pdf

[9] Muhammad Usman Hadi, et al., "Large Language Models: A Comprehensive Survey of Their Applications, Challenges, Limitations, and Future Prospects," TechRxiv, 2025. Available: https://www.techrxiv.org/users/618307/articles/682263-large-language-models-a-comprehensive-survey-of-its-applications-challenges-limitations-and-future-prospects

[10]Yejin Bang, et al., "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," arXiv, 2023. Available: https://arxiv.org/pdf/2302.04023 G.

[11] Beeyani, “Efficiency behind the pass: A study on lean management practices in professional kitchens,” *Sarcouncil Journal of Humanities and Cultural Studies*, vol. 4, no. 3, pp. 55–62, 2025.

[12]P. A. Mintah, “Self-financed real estate growth models for sustainable development,” *Journal of International Crisis and Risk Communication Research*, pp. 2565–2574, 2024, Available: https://doi.org/10.63278/jicrcr.vi.3424

[13]J. Boadi-Mensah, “Smart cities and sustainable waste systems: Innovations in urban solid waste management,” *Review of Contemporary Philosophy*, vol. 23, no. 2, pp. 7950–7959, 2024

[14]*Journal of Information Systems Engineering & Management*, vol. 8, no. 2, 2023. [Online]. Available: https://jisem-journal.com/index.php/journal/vol8_iss2