Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Architecting AI-Augmented Enterprise Software Systems: A Systematic Framework for Scalable, Secure, and Event-Driven Cloud-Native Applications

Garima Agarwal
Software Developer, Pyramid Consulting Inc, Alpharetta, GA, USA
Corresponding Author: Garima Agarwal, **E-mail**: agarwal24garima@gmail.com

| ABSTRACT

The increasing incorporation of artificial intelligence into software systems in various enterprises has further complicated architectural concerns related to scaling, security, governance, and adaptability. Although cloud-based and event-based architectural approaches are promising for modern enterprise systems, existing architectural approaches rarely consider artificial intelligence incorporation, cloud-based systems, and event-based systems as integral parts. These further limits the potential for artificial intelligence-based decision-making systems in various enterprises. This paper proposes a systematic architectural framework for AI-augmented enterprise software systems, which combines cloud-native design principles and event-driven coordination. By using a design science research approach, a systematic approach is followed in developing the architecture using synthesized architectural requirements from enterprise systems, AI service lifecycles, and distributed cloud environments. In addition, a layered abstraction is introduced in the proposed architecture for enterprise software systems, where AI services are treated as first-class components. The framework is validated using analytic, quality attribute-driven architectural reasoning, thus verifying its ability to tackle key concerns of an enterprise, such as scalability with varying AI-driven workload, secure interactions between services, resistance to partial failures, and long-term maintainability. By abstracting architectural patterns rather than tying them to implementation technologies, we provide a generalized blueprint for designing enterprise-grade AI systems. The proposed framework offers practical advice to enterprises looking to move from initial AI system deployment to scalable, secure, adaptable, and cloud-native systems.

| KEYWORDS

AI-augmented enterprise architecture, Cloud-native systems, Event-driven architecture, AI service lifecycle governance, Scalable and resilient software systems

| ARTICLE INFORMATION

ACCEPTED: 20 February 2026 **PUBLISHED:** 03 March 2026 **DOI:** 10.32996/jcsts.2026.8.5.2

1. Introduction

The architectural form of enterprise software systems is presently undergoing a major change as a result of the convergence of artificial intelligence, cloud computing, and distributed application technology. Modern enterprises rely significantly on the ability of artificial intelligence technology to provide efficient support for intelligent decision automation, predictive analytics, and adaptive process optimization. However, there exist several issues associated with the integration of artificial intelligence technology in the development of enterprise software systems. These issues have been found to be significantly amplified in a heterogeneous environment [1].

Cloud native architectures, as characterized by microservices, containerization, orchestration, and elastic provisioning of infrastructure, have emerged as a fundamental construct for scalable enterprise systems. At the same time, event-driven architectures have also emerged as a dominant construct due to their inherent capability to facilitate asynchronous communication, real-time response, and loose coupling of distributed services. Although both of these paradigms, on an individual basis, offer a number of benefits to modern software systems, the integration of these paradigms with AI-based services is still fragmented within the context of modern software architectures. Architectural constructs, as proposed earlier, have often focused on the deployment of AI services, cloud native architectures, and event-driven architectures as separate constructs, leading to a number of silos that limit the evolvability of modern software systems [2][3].

Additionally, the adoption of AI in enterprises also raises system-level concerns that go beyond performance-related aspects. Questions such as how to manage the lifecycle of the AI model, how to ensure the security of data access, how to manage decisions made by the AI system, and how to ensure that all the constraints of the enterprise are met, among others, make it necessary to ensure that the architecture of the enterprise system is not only systematic but also scalable. The absence of a unified architecture for AI-based enterprise systems raises uncertainty among enterprise architects and decision-makers [4].

To address this need, this research develops a comprehensive architectural framework for software systems in the context of AI technologies, such that the framework takes into account the requirements of cloud computing and event-driven software design. This research provides an important contribution to the design of software systems in the context of enterprise computing, considering the requirements of scalability, security, and decision-making that are intrinsic to software systems infused with AI technologies.

2. Conceptual Foundations and Related Work

2.1 Evolution of Enterprise Software Architecture

Enterprise software architectures have seen significant changes and evolution in the past two decades with the increase in system complexity, scale, and integration needs. Traditional monolithic architectures were based on centralized control and coupling, which were easy for development but restricted the system's ability to scale and adapt. Service-oriented architecture (SOA) introduced modularity and integration, but SOA-based systems often had fixed service contracts and centralized control, which restricted the ability of the system to adapt and change with the evolving needs of the enterprise. Microservices and cloud-native architectures have helped solve these issues by providing decentralized governance and scaling. Despite these advances in enterprise software architectures, the addition of AI components has brought new architectural challenges such as computational heterogeneity, workload dynamics, and data-centric processing, which are not within the scope of traditional enterprise architectures [5].

2.2 AI-Augmented Software Systems

The software systems enabled by AI are distinguished in a fundamental way from traditional systems based on rules and transactions, as they are based on probabilistic models and data-driven inference. Furthermore, the role of the AI components within the system can be as decision-support services, autonomous control agents, or predictive models integrated within the enterprise systems. The deployment strategy for the AI systems has been studied in the past, and the model-as-a-service and pipeline-based machine learning architecture are the most discussed ones. However, the deployment strategy for the AI systems often tends to focus more on the algorithmic and deployment efficiency, leaving the overall system-level considerations and trade-offs behind. The workloads of the AI systems are also non-deterministic, and the performance of the system is dynamic, which makes the architectural approach of the systems not suitable for the evolving workloads [6].

2.3 Cloud-Native and Event-Driven Architectural Paradigms

Cloud-native paradigms such as containerization and orchestration platforms, as well as serverless computing, are now an integral part of the scalability and efficiency of enterprise-level systems. Such paradigms allow for the fine-grained management of resources and the evolutionary development of services, making them suitable for the development of computationally intensive artificial intelligence services. Concurrently, event-driven architectures (EDA) allow for the development of asynchronous communication and reactive systems, allowing for the development of services that respond to events rather than requests. Previous studies have demonstrated the efficiency of using EDA in the development of distributed systems, allowing for the development of such systems to become more responsive, fault-tolerant, and scalable. However, the architectural development of artificial intelligence services using event-driven paradigms is an area that is not sufficiently explored, particularly in the development of feedback mechanisms, decision-driven events, and learning pipelines [7].

2.4 Synthesis and Research Gap

The existing body of literature contributes significantly to the development of perspectives regarding enterprise architecture modernization, AI system deployment, and cloud-native design principles. However, these research domains have been developed in a relatively isolated manner, and there is a lack of guidance regarding the integrated development of AI systems, cloud-native infrastructure, and event-driven communication at an architectural level. Furthermore, there is a lack of a unified framework that aligns the lifecycle of AI services with cloud-native scalability and event-driven coordination mechanisms. This lack of a unified framework has led to the development of a systematic architectural framework that moves beyond isolated design decisions regarding enterprise AI systems [8].

3. Research Methodology and Framework Development Approach

The research utilizes the design science research (DSR) research methodology for the development of a systematic architectural framework for AI-augmented enterprise software systems. This research methodology is particularly relevant for the research context because the primary purpose of the research is the development of an enterprise software system that faces the problem of relevance while providing conceptual knowledge. The research methodology does not emphasize the importance of performance evaluation.

The research process is organized into four distinct phases. Firstly, the research identifies the problem definition and motivation based on an exhaustive analysis of existing literature, which emphasizes the challenges faced in scaling, securing, and governing AI-based enterprise systems in distributed cloud environments. Secondly, the research identifies the architectural requirements based on the analysis of existing limitations in various enterprise, AI, and cloud architectures, with special emphasis on scalability, events, and governance. Thirdly, the framework design phase involves transforming the requirements into a layered architectural model that includes AI services, events, and cloud abstractions. The framework is technology-agnostic, which allows its application across various enterprise systems and cloud platforms.

To ensure conceptual validity, the proposed architecture is subject to logical and scenario-based architectural reasoning, which is an accepted software architecture research practice for evaluating software frameworks. This involves assessing how well the framework reacts and responds to various scenarios in an enterprise environment, such as spikes in AI inference requests, event-based decision automation for AI systems, and secure inter-service communication for AI systems. Finally, the framework is subject to various software architectural quality attributes such as scalability, security, resilience, and maintainability.

The outcome of this approach is the derivation of a generalizable architectural framework that incorporates AI system design, cloud native, and event-driven computing paradigms. The focus of the architectural principles, rather than implementation details, is the main contribution of the research to the derivation of a reusable foundation for the scholarly and practical use of AI-augmented software systems.

4. Architectural Requirements for AI-Augmented Enterprise Systems

The addition of artificial intelligence into enterprise software systems fundamentally changes the architectural requirements, which now extend beyond the traditional concerns of functional correctness and transactional performance. Enterprise software systems with added artificial intelligence require architecture to accommodate dynamic workloads and probability-driven decision-making. Architectural requirements for enterprise software systems with added artificial intelligence will now extend beyond traditional concerns. Four architectural requirement dimensions are synthesized as essential for enterprise software systems with added artificial intelligence.

4.1 Scalability and Elasticity

The computational characteristics of AI workloads are highly variable, and this is especially true for inference-centric enterprise use cases like real-time recommendation, anomaly detection, and predictive decision support systems. Therefore, there is a need for an enterprise architecture to facilitate horizontal scaling and elastic resource provisioning to meet variable demand without compromising system performance. However, unlike regular business services, AI-based inference services may require special resources and may have variable or bursty demand patterns. Therefore, there is a need for fine-grained scaling, isolation, and resource efficiency in distributed systems. This is not just limited to scaling compute resources; there is also a need to scale access to data, models, and events.

4.2 Security, Privacy, and Governance

Security and governance issues related to the augmentation of the enterprise with AI are not properly addressed within the context of existing security paradigms for the enterprise. There are issues related to the security and protection of sensitive data, models, and decision-making logic, as well as the need for traceability and accountability for decisions made within the context of the enterprise, which are related to the architectural level and not the operational level. Governance must be supported within the architecture and not be considered an operational issue.

4.3 Event-Driven Responsiveness and Adaptability

Enterprise AI systems are being used in environments that require real-time or near-real-time responses. Event-driven architectures support loosely coupled services in their ability to make asynchronous responses to changes in system state, external events, or insights generated by the AI system. Architectural support for event-driven communication between components can provide the means for AI components to trigger adaptive actions, coordinate distributed services, and provide the basis for creating feedback loops that support continuous learning. This is critical in maintaining the resilience of the system in the context of distributed enterprise ecosystems.

4.4 Maintainability and Evolvability

The evolution of AI models and enterprise requirements is ongoing, driven by changing data distributions, changing requirements, and constraints from regulators. As a consequence, the AI-augmented architecture must prioritize maintainability and evolvability. This includes requirements such as model versioning, the ability to update services independently, and the ability to deploy incrementally without disrupting the enterprise. Architectural modularity and coupling must be minimized to enable incremental evolution with minimal technical debt.

5. Proposed AI-Augmented Enterprise Architecture Framework

This section proposes an architectural system for AI-enhanced enterprise software systems, which considers cloud-native design principles and event-driven coordination mechanisms. This architectural system is designed to meet the architectural needs defined in Section 4, which are related to the integration of scalable AI, secure service interaction, and adaptive decision-making within distributed enterprise environments. Instead of depending on specific technologies, the proposed architectural system considers abstracted architectural layers and interaction patterns for its applicability in diverse industries and environments.

5.1 Layered Architectural Overview

The proposed framework is based upon an architectural model with layered structure for achieving better separation of concerns, modularity, and autonomous evolution of system components. At the highest level of abstraction, the architecture is divided into six logical layers: (i) the presentation layer, (ii) the application and API layer, (iii) the AI services layer, (iv) the event streaming and coordination layer, (v) the data and model management layer, and (vi) the cloud infrastructure layer.

The Presentation Layer offers user interfaces and external access points that abstract away the complexities associated with artificial intelligence from the user's perspective. The Application and API Layer is responsible for integrating business logic and facilitating interactions between various services and artificial intelligence technologies. The AI Services Layer represents encapsulated artificial intelligence inference, decision technologies, and intelligent adaptability. The Event Streaming Layer enables asynchronous interactions and reactivity in systems. Finally, the Data and Model Management Layer and Cloud Infrastructure Layer are responsible for governing data and model management and providing cloud-based scalable resources, networking, and monitoring.

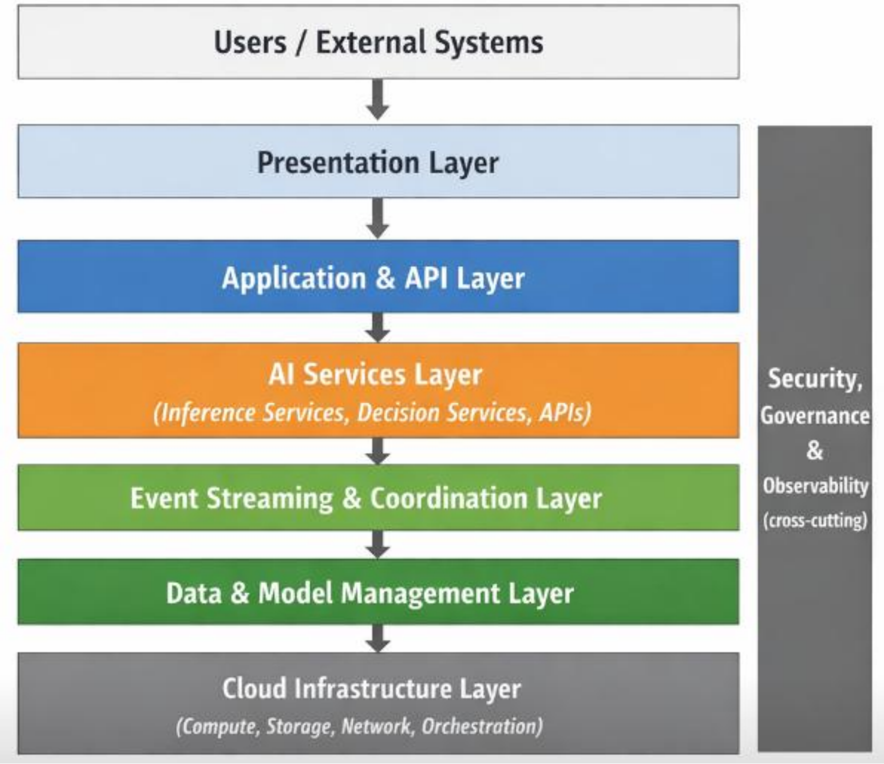
Users / External Systems
Presentation Layer
Application & API Layer
AI Services Layer
(Inference Services, Decision Services, APIs)
Event Streaming & Coordination Layer
Data & Model Management Layer
Cloud Infrastructure Layer
(Compute, Storage, Network, Orchestration)
Security, Governance & Observability (cross-cutting)

Fig 1. Layered AI-augmented enterprise architecture

The above figure shows a multilayered, AI-enhanced enterprise architecture, where user interactions and interfaces with external systems pass through the presentation and application/API layers, followed by a special AI services layer, and then an event streaming and coordination layer. After that, there is a data and model management layer, as well as a cloud infrastructure layer that provides scalable computing, storage, networking, and orchestration. However, as is clear from all of these layers, there is a cross-cutting capability that deals with security, governance, and observability, and that provides enterprise-grade functionality within a cloud-native, event-based world.

5.2 AI Services Layer Design

The AI Services Layer is a key element of the proposed architecture and aims to facilitate modularity, scalability, and governability of intelligence. A key principle of the architecture is to decouple the process of training the models from the inference services, thus allowing for independent scaling of these components. The process of training the models tends to be data-intensive and asynchronous, whereas the inference services are required to have low latency and high availability.

The components of inference in AI systems are designed as stateless services, which allows for scalability and fault tolerance in the system. Access to the models is controlled through standardized service interfaces, which allows enterprise systems to use the services of AI without being dependent on the specific algorithm used.

5.3 Event-Driven Integration and Adaptive Coordination

In order to achieve the goal of real-time response and decoupling integration, an event-driven integration layer is provided as a primary constituent of the framework's architecture. Instead of point-to-point synchronous interaction between enterprise services and AI components, there is an asynchronous interaction through event streams. Business events and AI-generated insights about system state changes are published as events and can be independently consumed and acted upon by consuming services.

Such event-driven coordination can be useful in creating flexible feedback loops where AI results trigger subsequent actions and eventually generate data for continuous learning processes. This pattern is useful in dynamic enterprise environments where conditions are constantly changing and the timing of actions is critical. The decoupled characteristics of event-based communication also make them useful for creating fault-tolerant systems.

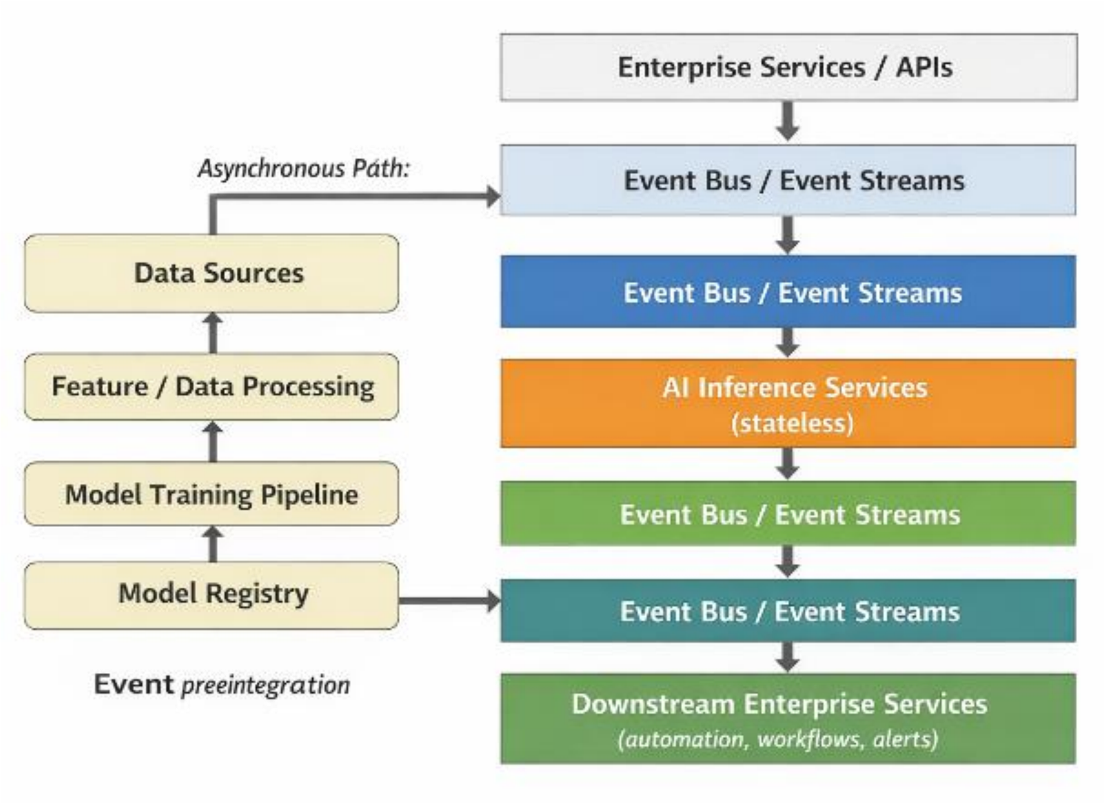
Enterprise Services / APIs
Asynchronous Páth:
Event Bus / Event Streams
Data Sources
Event Bus / Event Streams
Feature / Data Processing
AI Inference Services (stateless)
Model Training Pipeline
Event Bus / Event Streams
Model Registry
Event Bus / Event Streams
Event preeintegration
Downstream Enterprise Services (automation, workflows, alerts)

Fig 2. Event-driven integration architecture

The above figure depicts an event-driven integration architecture in which enterprise services communicate with each other using a central event bus to publish and consume events. These events then trigger stateless AI inference services, which are then propagated as events to other enterprise services to be automated, processed, or alerted. An asynchronous and decoupled training pipeline, which includes data sources, feature/data processing, model training, and a model registry, continuously trains the inference services used in the AI integration architecture.

5.4 Cloud-Native Infrastructure Enablement

The Cloud Infrastructure Layer is the foundation for the scalable, resilient, and observable architecture. Containerization and orchestration enable the deployment and management of services, and the scaling capabilities are for the varying AI workloads. Serverless computing can also be utilized for the burst-based inference or event handling, making the architecture more cost-effective without compromising the responsiveness.

Observability is a set of capabilities, including distributed tracing, monitoring, and logging, and is integrated across the layers. By aligning the AI services with cloud native operational best practices, the framework ensures that intelligence is not viewed as a separate, ancillary capability, but rather as a key, controllable part of an enterprise software system.

6. Security and Governance Architecture for AI-Augmented Systems

Security and governance are two primary considerations for enterprise software systems that incorporate AI, considering the significant role that intelligence plays in the automation of business decisions and operations. However, unlike other enterprise software systems, there are additional security concerns that arise when artificial intelligence is incorporated, such as the use of the model, the leakage of the model, the lack of transparency in the model, and the model drift.

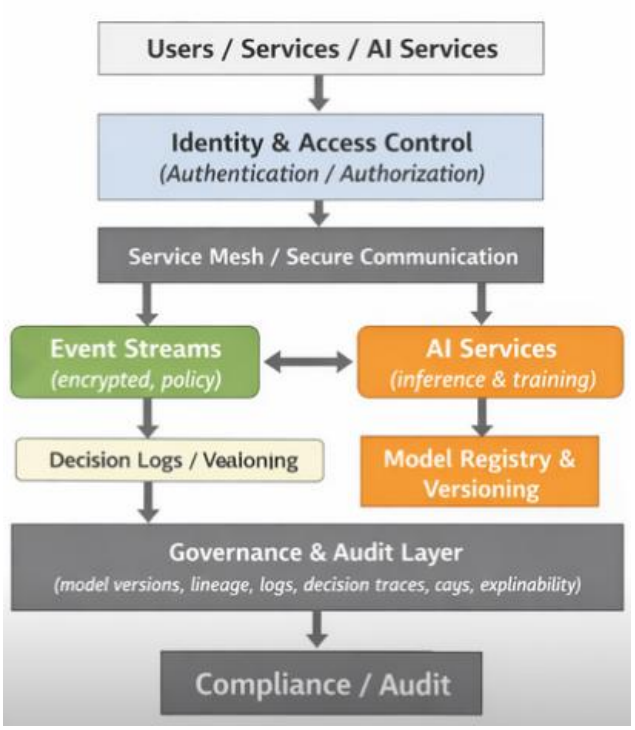
Users / Services / AI Services
Identity & Access Control
(Authentication / Authorization)
Service Mesh / Secure Communication
Event Streams
(encrypted, policy)
AI Services
(inference & training)
Decision Logs / Vealoning
Model Registry & Versioning
Governance & Audit Layer
(model versions, lineage, logs, decision traces, cays, explinability)
Compliance / Audit

Fig 3. security and governance architecture

The above figure describes a security and governance model for AI-enabled enterprise systems, where all users, enterprise services, and AI services are authenticated and authorized using identity and access control before any kind of communication takes place through a secure service mesh. The event streams, as well as AI services that include both inference and training, are exchanged in a bidirectional manner, while a model registry provides support for model management using a versioning system, enabling traceability, lineage, explanations, and accountability using a centralized governance and audit layer, which provides support for decision logs.

6.1 Zero-Trust and Identity-Centric Security Design

The proposed framework will utilize a zero-trust security model in which no service, user, or AI is trusted implicitly. The interactions between enterprise services, AI inference services, and events will be managed through identity-aware authentication and authorization mechanisms. Service-to-service communication will be achieved through strong identity verification, access controls, and encrypted communication, thereby allowing AI services to run exclusively in authorized contexts. This is particularly important in a distributed cloud-native environment, in which scaling can increase the system's attack surface.

6.2 AI Model Governance and Decision Accountability

AI governance is distinguished from software governance by the probabilistic nature of the decisions made by the machine learning model. As such, the architectural framework includes the lifecycle governance of the model as a foundational concern, which includes version control, lineage, and the ability to track the inputs and outputs of the model. This is essential for the auditability and analysis of the decisions made by the AI, which is critical for the enterprise to meet the requirements for compliance with the law. In addition, the architectural provisions for explainability allow the stakeholders within the enterprise to investigate the decisions made by the AI without the details of the model's inner workings.

6.3 Secure Data and Event Stream Management

In the context of event-driven coordination, there are particular security issues to address, since the information carried by the events can include sensitive business information or AI-derived insights that involve different services. In the framework, there are security controls defined at the level of the event streams, which include encryption, segmentation, and policy-based event consumption. In this way, the permissions to publish, subscribe, and replay the events are controlled, aligning with the data governance requirements of the enterprise.

7. Architectural Evaluation and Quality Attribute Analysis

The proposed AI-augmented enterprise architecture is evaluated through a quality attribute-driven architectural analysis, which is a recognized technique for assessing complex software systems in situations without empirical deployment. Instead of using benchmarking techniques for specific implementation parameters, the evaluation focuses on the extent to which the architecture satisfies major enterprise quality attributes.

7.1 Scalability and Performance

Scalability is achieved by decoupling AI inference services, event-driven coordination, as well as elastic cloud-native infrastructure. With stateless AI services, horizontal scaling can be achieved in response to varying demands. On the other hand, asynchronous event processing eliminates latency amplification in response to increased demands. With compute-intensive AI services decoupled from transactional services, resource contention is minimized, thus maximizing system throughput. Compared with synchronous, coupled architectures, the proposed framework has better scalability attributes for burst-driven, data-intensive, as well as enterprise environments.

7.2 Security and Resilience

From a security perspective, the integration of zero trust principles and identity-centric access control enables the security of service interactions in a dynamically scaled and distributed environment. Event stream governance also addresses the risk of unauthorized data propagation. In terms of resiliency, the pattern prevents cascading failures through decoupling service availability from real-time dependencies. Fault isolation at the service and event stream levels also improves resiliency in the presence of partial failures or degraded AI components.

7.3 Maintainability and Evolvability

The layered and modular design of this framework promotes maintenance by providing the flexibility to modify individual components of artificial intelligence, business logic, and infrastructure. This facilitates the development of enterprise workflows while incorporating improvements to artificial intelligence. Compared to other monolithic or orchestrated systems, this suggested architecture effectively addresses the issue of technical debt.

7.4 Trade-Off Analysis

This architecture also improves scalability, flexibility, and governance. However, there is also added complexity in terms of orchestration and observability. This is an intrinsic aspect of distributed and event-driven systems and architecture. However, the benefits of flexibility, resilience, and enterprise readiness outweigh the added complexity of the architecture for AI-driven applications.

8. Managerial and Enterprise Implications

The proposed architecture of the AI-augmented enterprise architecture holds significant implications for managerial decision-making processes. The framework provides a blueprint for integrating AI services into the existing architecture of the organization, which is important for enterprise architects and technology leaders. It allows organizations to take incremental steps towards the adoption of AI services while minimizing disruptions to the existing architecture of the organization.

From a managerial point of view, the framework enables better-informed investment decisions through the identification of the architectural preconditions for the sustainable adoption of AI. Organizations are not encouraged to implement isolated initiatives for the development of AI capabilities; instead, they are encouraged to align the development of their AI capabilities with their cloud-native infrastructure and governance structures, which reduces the long-term operational risks and increases the organization’s ability to scale its capabilities for multiple business domains.

It also has implications for organizational readiness, where managers must recognize that the application of AI-enabled, event-driven architectures requires cross-functional collaboration between software engineers, data scientists, security specialists, etc. Hence, managers must focus on developing architectural literacy, DevOps maturity, etc., to realize the full potential of AI integration. Furthermore, by explicitly considering security and model governance, managers can be confident that their organization is meeting current and emerging regulations, thus using AI in a responsible manner while also ensuring stakeholder trust.

In conclusion, the proposed framework allows enterprises to move from using AI in an exploratory manner to using AI in a systematically engineered, enterprise-class manner, thus positioning them for continued competitiveness in a digital world.

9. Limitations and Future Research Directions

The research offers a systematic architectural framework for AI-augmented enterprise software systems, with several limitations acknowledged. First, the proposed framework is analytically validated, rather than empirically validated. Although the proposed design science and scenario-based reasoning approaches are widely accepted research methodologies for software architecture research, the proposed framework can benefit from empirical validation. Second, the proposed software architecture is technology-neutral, which is useful for its applicability to the enterprise domain, although it precludes the evaluation of the trade-offs in terms of the technology used.

These limitations provide several avenues for future work. For instance, empirical work in different domains can be carried out to validate the framework under operational constraints. Quantitative evaluation of the EDAI architectures against traditional synchronous-based approaches can provide further insights into the performance benefits of EDAI architectures. In addition, future work can investigate the incorporation of human-centric design principles, explainable AI, and ethics to align the AI-infused enterprise systems with the emerging Industry 5.0 and responsible AI paradigms.

10. Conclusion

The purpose of this research is to explore the increasing architectural challenges of integrating artificial intelligence technologies into enterprise software systems. With the increasing influence of artificial intelligence on enterprise decision-making, the challenges of accommodating the scalability and flexibility of enterprise software systems are being faced. By combining the evolutionary perspectives of enterprise architecture, cloud computing, and event-driven software design, the analysis of the topic reveals the significant gap in the literature on the need for a unifying framework for the integration of artificial intelligence services in modern enterprise software platforms.

In order to bridge this gap, the research presents a framework for a structured AI-augmented enterprise architecture that includes aspects of layered abstraction, cloud-native scalability, and event-driven coordination. The architecture includes the explicit consideration of aspects of modularization of AI services, asynchronous interaction between services, and security/governance aspects. The architecture is evaluated based on various aspects of quality attributes, which prove its support for key aspects of enterprise requirements, including scalability, partial failure resilience, secure inter-service communication, and long-term maintainability. The most important aspect is that the framework includes AI as a primary aspect, embedded at all levels of the architecture, including the infrastructure, data, and governance layers.

This study goes beyond theoretical analysis of architectural design by providing practical guidelines for enterprises that seek to move beyond experimental AI deployment towards developing intelligent systems. By providing a generalized blueprint, the framework enables enterprises to make informed decisions in architectural design while allowing for flexibility in adapting to new AI technologies. With enterprises becoming increasingly dependent on AI-based automation tools, the framework provides a basic guideline for developing robust enterprise software systems that can sustainably support digital transformation.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

1. Patterson L, Pigorovsky D, Dempsey B, et al (2022) HiveMind: A Hardware-Sofware System Stack for Serverless Edge Swarms. In: Proceedings - International Symposium on Computer Architecture
2. Vankayalapati RK (2025) Architectural foundations of hybrid cloud solutions. In: Deep Science Publishing
3. Moreschini S, Pour S, Lanese I, et al (2025) AI Techniques in the Microservices Life-Cycle: a Systematic Mapping Study. Computing 107:. https://doi.org/10.1007/s00607-025-01432-z
4. Moin A, Badii A, Gunnemann S, Challenger M (2025) Enhancing Architecture Frameworks by Including Modern Stakeholders and Their Views/Viewpoints. In: Proceedings - 2025 8th International Conference on Information and Computer Technologies, ICICT 2025
5. Karimi M, Barforoush AA (2023) Proposing a Dynamic Executive Microservices Architecture Model for AI Systems. ArXiv
6. Feng J, Lansford JL, Katsoulakis MA, Vlachos DG (2020) Explainable and trustworthy artificial intelligence for correctable modeling in chemical sciences. Sci Adv 6:. https://doi.org/10.1126/sciadv.abc3204
7. Lin J, Xie D, Huang J, et al (2022) A multi-dimensional extensible cloud-native service stack for enterprises. Journal of Cloud Computing 11:. https://doi.org/10.1186/s13677-022-00366-7
8. Sinha S, Lee YM (2024) Challenges with developing and deploying AI models and applications in industrial systems. Discover Artificial Intelligence 4:. https://doi.org/10.1007/s44163-024-00151-2