IJC
International Journal of Computer (IJC)
Peer-Reviewed International Journal
ISSN: 2307-4523
https://ijcjournal.org

Layered Architecture in Enterprise BPM Systems as a Foundation for Long-Term Extensibility

Mykhailo Krasovskyi*

Lead Solution Architect at Trinetix, Kyiv, Ukraine

* *Corresponding author: mkrasovskiy85@gmail.com*

Accepted: 22/07/2026 | Published: 5/09/2026

Abstract

This article addresses the challenge of ensuring the long-term extensibility of enterprise business process management (BPM) systems under conditions of increasing architectural complexity and a growing number of integrations with external systems. Particular attention is paid to the limitations of traditional architectural approaches based on tightly coupled components, where even minor modifications affect multiple system layers and reduce maintainability. The study combines an analytical review of recent research on enterprise architecture, business process management, event-driven execution, and data-driven BPM with a case analysis of the OpenText MBPM platform. The findings demonstrate that a system's capacity for sustainable evolution is determined primarily by its architectural structure and the distribution of changes across layers rather than by the number of implemented functions or modular extensions. Based on the analysis, the study proposes the Layered Extensibility Architecture for BPM Systems (LEA-BPM), which distinguishes a stable architectural core comprising data storage, process execution, and deployment from design and interaction layers where modifications are localized. The proposed model extends existing approaches to BPM architecture by conceptualizing extensibility as a structural property achieved through change isolation across architectural layers. The practical value of the model lies in supporting the design and evolution of enterprise BPM systems operating in environments characterized by continuous process changes and intensive system integration, enabling the introduction of new functionality and external connections without modifying core system components. The article is intended for researchers and practitioners in enterprise architecture, enterprise information systems development, business process management, and digital transformation.

Keywords: business process management systems, enterprise architecture, layered architecture, extensibility, system integration, architectural coupling, enterprise information systems.

1. Introduction

Enterprise business process management (BPM) systems have become a central component of the modern digital enterprise. They support document approval, task management, cross-departmental collaboration, and integration with external systems [1]. As transaction volumes continue to grow, organizational processes become more complex, and architectural workloads increase accordingly. Architectural solutions based on monolithic application principles rapidly approach their scalability

limits. Even minor modifications affect multiple architectural layers, interface adjustments require corresponding changes in business logic, and the integration of external services influences the data storage layer. As a result, component coupling intensifies and system maintainability declines [2].

The increasing frequency of process changes further amplifies the need for architectural modifications. Business regulations are continuously revised, process scenarios evolve, and participant structures change over time. New integrations introduce additional interaction points, increasing architectural complexity. Processes involving document management and regulatory compliance impose further constraints related to version control, status management, and retention policies. In professional practice, extensibility is often interpreted merely as the availability of development tools or extension mechanisms, whereas the fundamental distinction between sustainable and overloaded systems lies in their underlying architectural structure [3]. Although recent studies have extensively examined enterprise architecture, modularity, and BPM system evolution, the distribution of architectural changes across system layers has received considerably less attention. Existing approaches predominantly focus on modular extensions, while the architectural mechanisms ensuring long-term evolvability remain insufficiently conceptualized.

The aim of this study is to develop and theoretically substantiate a layered architecture model for business process management systems that supports long-term evolution through the separation of a stable architectural core from adaptable layers. To achieve this objective, the study addresses the following tasks:

– to analyze approaches to layered architecture within enterprise architecture and business process management systems;
– to identify the limitations that arise when systems with highly coupled components evolve;
– to examine the architecture of an enterprise business process management system based on an applied case study;
– to formulate principles for designing an architecture oriented toward long-term development.

The research hypothesis is that the long-term evolution of enterprise business process management systems is determined by the nature of their architectural design, whereby the separation of stable execution layers from adaptable interaction layers localizes changes within a limited portion of the system and preserves overall architectural integrity as new functionality is introduced.

The scientific novelty lies in proposing that the widely adopted modular approach to the extensibility of business process management systems is insufficient to ensure their long-term evolution because it does not eliminate architectural coupling between system components. The study shows that sustainable development depends primarily on the structural separation of architectural layers, with changes isolated outside the execution core rather than accommodated through the continuous addition of functional modules. On this basis, the paper proposes an original layered architecture model in which extensibility is achieved through the distribution of changes across architectural layers and their localization within the upper layers of the system.

2. Materials and Methods

This study was conducted as an analytical review of the scientific literature followed by the development of a conceptual layered architecture model for BPM systems designed to support long-term evolution. The analysis focused on studies addressing enterprise system architecture, layered architectural models, business process management, application integration, event-driven process execution, and the architectural factors influencing system evolution and adaptability.

The literature search was performed using the Google Scholar, SpringerLink, and MDPI databases. The following keywords and their combinations were used: enterprise architecture, business process management system, layered architecture, modularity, system integration, event-driven process

execution, component coupling, architectural complexity, system extensibility, process lifecycle, digital transformation, and enterprise information systems. Boolean operators (AND and OR) were applied to refine the search. The review included publications published between 2022 and 2025 that presented analytical or conceptual findings related to the architecture and evolution of business process management systems. Studies focusing exclusively on narrow technical implementations or individual software tools were excluded from further analysis.

An initial set of 45 publications was identified. Following an assessment of their relevance to the research objectives, 15 sources were selected for the final review. The analysis showed that contemporary research primarily emphasizes modularity, system integration, architectural quality, and process lifecycle management. In contrast, relatively little attention has been devoted to the distribution of architectural changes across system layers and the mechanisms for isolating those changes. This research gap provided the foundation for developing the proposed layered architecture model.

The study also employed a case study approach to examine the architecture of an enterprise business process management system comprising a data storage layer, a process execution layer, a deployment layer, and separate design and interaction layers responsible for user and external system communication. This made it possible to compare theoretical concepts with practical architectural solutions and to identify stable design principles that support long-term system evolution.

Several limitations of the study should be considered. First, Layered Extensibility Architecture for BPM Systems (LEA-BPM) is proposed as a conceptual architectural model and has not yet been quantitatively validated across multiple BPM platforms. Second, the applied case analysis is based on a single enterprise BPM platform, which limits the direct generalization of the identified architectural patterns to systems based on different execution and integration mechanisms. Third, the study does not quantify change propagation, component coupling, modification effort, deployment complexity, or maintenance cost. Finally, the literature review was limited to publications from 2022 to 2025 and was intended to identify contemporary architectural patterns rather than provide an exhaustive systematic review. These limitations define the scope within which the findings and the proposed model should be interpreted.

3. Results

The analysis of architectural approaches to business process management systems reveals a consistent pattern. The long-term evolution of such systems is determined primarily by the structure of their architectural layers, whereas the set of implemented functions has a comparatively smaller influence. A comparison of enterprise architecture frameworks and BPM models demonstrates a multilayered organization in which business processes, applications, and infrastructure constitute interconnected layers characterized by different rates of change [4]. Within this perspective, extensibility is understood as a property of the system's internal architecture rather than the result of the number of available functions.

The integration of BPM with enterprise architecture reveals a clear internal hierarchy. The business layer contains processes, organizational roles, and business events. The application layer supports process execution through services, while the technology layer provides the underlying infrastructure and operational constraints. At the strategic level, business processes are aligned with organizational objectives and long-term development priorities [5]. The allocation of responsibilities across these layers establishes the fundamental logic governing how architectural changes propagate throughout the system.

Business processes evolve continuously, but changes to process models do not, by themselves, require modifications to the underlying BPM platform. Architectural changes become necessary only when new or updated business processes require additional functionality, new integrations, or

modifications to existing interaction mechanisms. In highly coupled architectures, such changes may propagate across multiple system components, increasing maintenance complexity and reducing architectural stability [6]. In contrast, layered architectures localize these modifications within the relevant layers and prevent their unnecessary propagation across the system. Event-driven process execution follows the same architectural principle, with process models, execution rules, and runtime states remaining logically separated while interacting through formalized mechanisms [7].

The architecture of the analyzed enterprise BPM system exhibits the same structural characteristics. The system is organized into five architectural layers. The lower part consists of the data storage, process execution, and deployment layers, which together form a stable architectural core. The upper part includes the process design and user interaction layers, where the majority of modifications are introduced. Consequently, the principal architectural boundary separates process execution from process configuration and adaptation [8].

These findings indicate that extensibility is primarily determined by architectural layering rather than by the availability of development tools. Software tools serve a supporting role, whereas sustainable evolution is achieved by restricting modifications to the upper architectural layers while preserving the integrity of the execution core. Such behavior is embedded in the architectural structure from the outset rather than emerging from subsequent functional expansion.

Additional evidence is provided by the analysis of data-driven BPM architectures [9]. In these approaches, the system is organized as a sequence of architectural layers, each introducing new capabilities without restructuring the underlying foundation. Progression across the layers is associated with the accumulation of data and increasingly sophisticated analytical processing, while the lower layers remain structurally stable. Table 1 summarizes the layers of data-driven BPM and their respective contributions to overall system architecture.

Table 1. Layers of Data-Driven BPM and Their Contribution to Architectural Extensibility (Compiled by the author based on source: [10])

Data-Driven BPM Layer	Position in Pyramid	Number of Capabilities / System Types	Share, %	Cumulative Share, %
Descriptive Process Analytics	1	4	40.0	40.0
Predictive Process Analytics	2	2	20.0	60.0
Prescriptive Process Optimization	3	2	20.0	80.0
Autonomous Process Execution	4	2	20.0	100.0

Although the original study addresses the maturity of data-driven BPM rather than extensibility itself, its layered progression is relevant to the present analysis because it illustrates how additional capabilities can be introduced while previously established architectural functions remain available. LEA-BPM applies the same general logic to architectural change rather than capability maturity: the central issue is whether new functionality can be introduced without requiring corresponding structural modification of lower execution layers. Each successive architectural layer introduces additional capabilities while preserving the functionality of the existing mechanisms. The same principle can be observed in practical enterprise architectures. The implementation of new functionality or the integration of an external service affects only the upper portion of the system, whereas the lower layers continue performing their responsibilities without modification. This separation of responsibilities across architectural layers establishes clear boundaries for change and preserves the overall stability of the system during its evolution.

The analysis also indicates that research attention across the BPM lifecycle is concentrated primarily on process initiation and initial configuration. Most efforts are devoted to process modeling and

implementation, whereas considerably less emphasis is placed on subsequent modification, adaptation, and long-term evolution. As a result, business process management systems generally provide effective support for process deployment but encounter increasing architectural complexity and a growing number of dependencies as they evolve over time. This imbalance constitutes an inherent architectural limitation. Table 2 presents the distribution of research activity across the major phases of the business process lifecycle.

Table 2. Distribution of Research Attention Across the BPM Lifecycle Phases (Compiled by the author based on source: [11])

BPM Lifecycle Phase	Share of Studies, %	Rank	Gap from Leader, p.p.
Modeling	35.4	1	0.0
Implementation	31.6	2	3.8
Monitoring and Mining	16.5	3	18.9
Analysis	7.0	4	28.4
Redesign	5.7	5	29.7
Identification	3.8	6	31.6

The distribution of research attention across the BPM lifecycle indicates that modeling and implementation receive substantially greater attention than redesign and subsequent adaptation. For the present study, this imbalance is important because long-term extensibility becomes relevant primarily after initial deployment, when existing processes, integrations, and interaction mechanisms must be modified without destabilizing the platform. The limited attention to redesign therefore reinforces the need to examine how BPM architectures accommodate change after their initial implementation. Changes to existing process models can typically be handled through process configuration and do not require modifications to the underlying BPM platform. Architectural changes become necessary during BPM system implementation when new functionality is introduced, such as integration with an external system. In highly coupled architectures, the addition of such functionality may affect multiple components and create chains of architectural dependencies.

In the analyzed OpenText MBPM platform architecture, this risk is addressed through the explicit separation of architectural layers. The data storage, process execution, and deployment layers operate as an integrated architectural foundation. Their interaction is coordinated through a unified data access and processing mechanism, which preserves consistency across the system while providing a stable basis for process execution [12].

During BPM system implementation, a new integration is introduced through the upper architectural layers. Its configuration is defined at the process design level, while the user interaction layer provides communication with the external system [13]. The core architectural components do not require corresponding modification: the process execution, data storage, and deployment mechanisms retain their existing structure, thereby preserving the stability of the execution environment.

4. Discussion

The results indicate that the critical architectural factor for long-term extensibility is not simply modularity, but the scope of change propagation within the system. In the analyzed architecture, routine

changes associated with process configuration, business scenarios, and external integrations are concentrated primarily in the design and interaction layers, while execution-oriented layers remain comparatively stable. This does not imply that the lower layers are immutable. Changes in persistence technology, runtime requirements, security mechanisms, or deployment infrastructure may also require modification of the architectural core. The relevant distinction is therefore between routine functional evolution, which can be localized, and foundational platform changes, which may affect the core architecture.

Architecture influences system evolution through the allocation of responsibilities among layers [14]. Each layer performs a defined set of functions and communicates with adjacent layers through stable interfaces, limiting the effects of local modifications on the rest of the system.

Stable interfaces provide the mechanism for this containment. They make communication between layers predictable and preserve it when other parts of the system change. Integrating a new external service can therefore be handled within the relevant layer without restructuring the entire architecture [15]. In this way, architectural boundaries control the spread of change and prevent unnecessary dependencies from accumulating as the system evolves.

This interpretation both corresponds to and differs from previous architectural research. Studies of enterprise architecture emphasize the separation and coordination of business, application, and technology concerns and the importance of maintaining alignment as enterprise systems evolve [1], [3], [5]. Research on process enactment further demonstrates the value of separating execution mechanisms from higher-level process representations [7], while studies of migration toward microservices and low-code environments examine modular decomposition and the reduction of implementation barriers [2], [14]. These approaches support the broader principle of architectural separation, but their primary focus is not the location and propagation of change within an enterprise BPM system. LEA-BPM develops this aspect further by treating change containment itself as a mechanism of extensibility. From this perspective, the existence of modules or extension tools is insufficient if the introduction of new functionality still requires coordinated modification across storage, execution, deployment, and interaction components.

Layer separation produces a specific pattern of development. New capabilities enter through the upper layers, while the core execution mechanisms continue to operate unchanged. Continuous development is therefore accommodated within the architecture itself, with stability maintained by directing modifications away from the system core.

The enterprise BPM system under analysis applies this principle through a sequence of clearly differentiated layers. Data storage, process execution, and deployment form a stable foundation that supports the reliable execution of business processes. Process design and user interaction are assigned to the upper layers. The distinction between these groups of layers reflects their different rates and sources of change. Process design and interaction layers respond directly to evolving business requirements, user scenarios, and external systems. Data storage, process execution, and deployment instead provide the mechanisms required for persistent state management, runtime execution, and operational delivery. They may therefore remain stable across multiple changes in business functionality, provided that those changes do not alter the fundamental technical requirements of the platform.

New external services are integrated according to the same arrangement. Their configuration can be introduced primarily within the process design and interaction layers, while the existing data storage, process execution, and deployment mechanisms can remain unchanged when the integration does not introduce new persistence, runtime, security, or deployment requirements. The distinction between routine functional evolution and changes affecting the execution foundation is thereby preserved.

By containing modifications within designated layers, this architecture limits their effects on the system as a whole. Functional growth leaves the architectural foundation intact, making extensibility a property of the design itself. When a new integration is added, the layered structure already determines where the required changes will occur and protects the core components from their impact.

The reviewed architectural approaches provide the basis for a model that treats extensibility as an intrinsic architectural property. Figure 1 presents the proposed LEA-BPM.

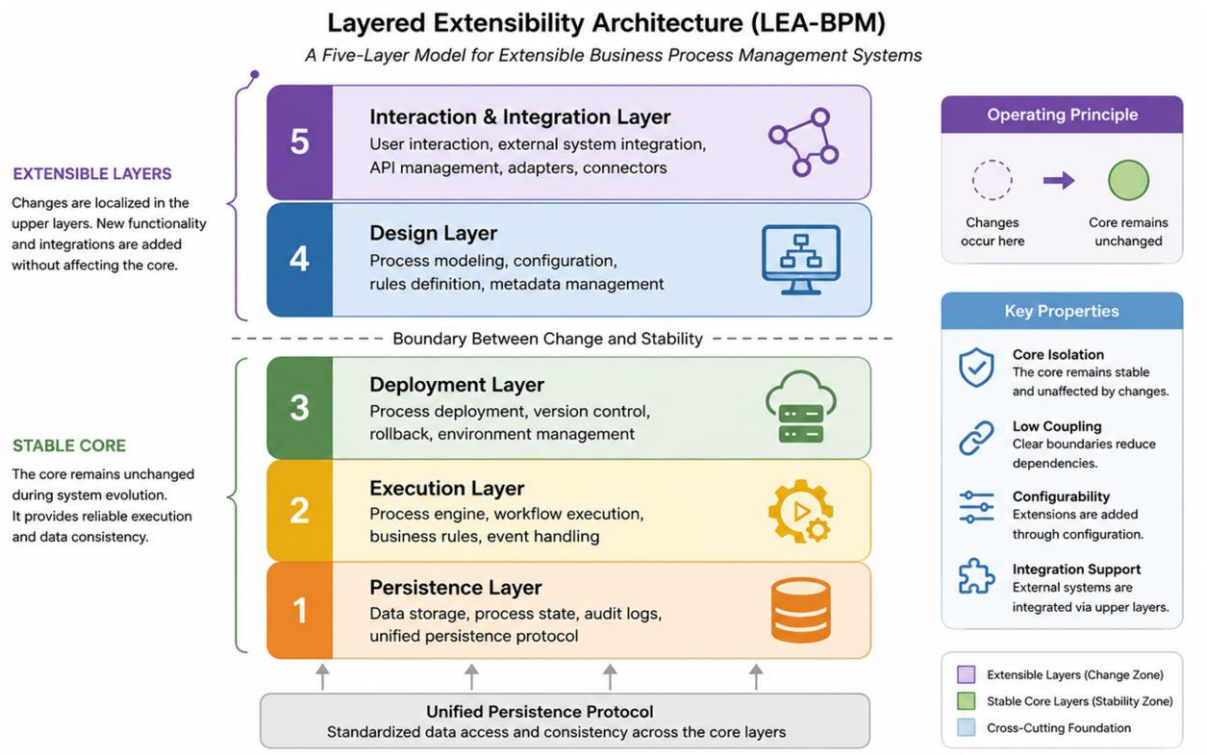


Figure 1. Layered Architecture for BPM System Extensibility (Author's elaboration)

The proposed model distinguishes the system into a stable architectural foundation and a dedicated domain of change. The lower architectural layers comprise data storage, process execution, and deployment mechanisms. Together, these components constitute the architectural core of the system, whose behavior remains stable as new functionality is introduced. Data management, process execution, and deployment are coordinated within a unified architectural framework that ensures consistent system operation.

The defining characteristic of the proposed model is the isolation of architectural changes. LEA-BPM does not treat layered architecture itself as a new architectural concept. Its contribution lies in interpreting the distribution and containment of change across layers as the principal mechanism of BPM system extensibility. Under this model, routine functional modifications are directed toward the upper layers, while component coupling is limited through the separation of architectural responsibilities. Extensibility is therefore associated not only with the availability of modules or development tools, but with the ability of the architecture to prevent local changes from propagating unnecessarily into the execution core.

This architectural approach fundamentally changes the pattern of system evolution. Functional growth no longer leads to increasing complexity within the architectural foundation. Instead, the system maintains its manageability as the number of external integrations expands. The layered architecture preserves an effective balance between architectural stability and continuous evolution.

The proposed model elevates the discussion from individual architectural solutions to general

architectural principles governing long-term system development. System behavior under changing conditions is predetermined by the architectural structure, while subsequent evolution occurs within clearly defined architectural boundaries. As a result, extensibility becomes an integral characteristic of architectural design rather than a consequence of incremental functional expansion.

5. Conclusion

The long-term evolution of enterprise business process management systems depends primarily on their architectural organization. System extensibility is shaped by the distribution of architectural changes and by the layers that absorb their effects. In highly coupled architectures, functional expansion increases structural complexity and makes the system more difficult to maintain. Layered architecture follows a different evolutionary pattern: it confines changes to a limited architectural domain while preserving the integrity of the system foundation.

This study develops the LEA-BPM, which conceptualizes system evolution as the controlled distribution of changes across architectural layers. The model separates a stable core comprising the data storage, process execution, and deployment layers from the higher-level layers responsible for process design and interaction with users and external systems. Core execution mechanisms remain stable, while architectural modifications are concentrated in the upper layers. Conventional modular approaches treat extensibility as the incremental addition of functional modules; LEA-BPM instead explains it through the architectural isolation of change.

The proposed architecture supports the design and continued development of enterprise BPM systems in environments where processes change frequently and systems require intensive integration. Layered separation combined with stable interaction mechanisms can allow new integrations and business scenarios to be introduced without corresponding changes to foundational components. By reducing the propagation of routine modifications across architectural layers, this structure is intended to support maintainability and preserve the system's capacity to adapt over time.

The findings support the research hypothesis within the scope of the literature review and the analyzed case. Sustainable evolution in enterprise BPM systems can be supported by the separation of a comparatively stable execution core from the architectural domain in which routine functional changes occur. Further cross-platform validation is required to determine how consistently this pattern applies to BPM systems based on different execution and integration architectures.

References

[1] E. Atencio, M. Mancini, and G. Bustos, "Ontology-based integration of enterprise architecture and project management: A systems thinking approach for project-based organizations in the architecture, engineering, and construction sector," Systems, vol. 13, no. 6, p. 477, 2025, doi: 10.3390/systems13060477.

[2] I. Oumoussa and R. Saidi, "The ontology-based mapping of microservice identification approaches: A systematic study of migration strategies from monolithic to microservice architectures," Computers, vol. 14, no. 4, p. 133, 2025, doi: 10.3390/computers14040133.

[3] E. Atencio, G. Bustos, and M. Mancini, "Enterprise architecture approach for project management and project-based organizations: A review," Sustainability, vol. 14, no. 16, p. 9801, 2022, doi: 10.3390/su14169801.

[4] M. Rosemann, W. M. P. van der Aalst, A. Van Looy, et al., "Business process management in the age of AI: Three essential drifts," Information Systems and e-Business Management, vol. 22, pp. 415–429, 2024, doi: 10.1007/s10257-024-00689-9.

[5] G. Fuentes-Quijada, F. Ruiz-González, and A. Caro, "Enterprise architecture and IT governance to support the BizDevOps approach: A systematic mapping study," Information Systems Frontiers, vol. 27, pp. 865–888, 2025, doi: 10.1007/s10796-024-10473-2.

[6] M. Saeedi Nikoo, S. Kochanthara, Ö. Babur, et al., "An empirical study of business process models and model clones on GitHub," Empirical Software Engineering, vol. 30, p. 48, 2025, doi: 10.1007/s10664-024-10584-z.

[7] A. Awad, F. Awaysheh, and H. López, "BEST: A unified business process enactment via streams and tables

for service computing," Process Science, vol. 2, p. 23, 2025, doi: 10.1007/s44311-025-00030-8.
[8] G. Pisoni and M. Moloney, "Responsible AI-based business process management and improvement," Discover Information Systems and Operations, vol. 3, p. 23, 2024, doi: 10.1007/s44206-024-00105-2.
[9] T. Skouti, R. Seiger, F. J. Furrer, et al., "RBPMN: The value of roles for business process modeling," Software and Systems Modeling, vol. 23, pp. 1375–1406, 2024, doi: 10.1007/s10270-024-01202-z.
[10] D. Chapela-Campa and M. Dumas, "From process mining to augmented process execution," Software and Systems Modeling, vol. 22, pp. 1977–1986, 2023, doi: 10.1007/s10270-023-01132-2.
[11] F. De Luzi, F. Leotta, A. Marrella, et al., "On the interplay between business process management and Internet-of-Things," Business & Information Systems Engineering, vol. 67, pp. 265–288, 2025, doi: 10.1007/s12599-024-00859-6.
[12] B. Uyanık and A. Sayar, "Developing web-based process management with automatic code generation," Applied Sciences, vol. 13, no. 21, p. 11737, 2023, doi: 10.3390/app132111737.
[13] T. Kampik, C. Warmuth, A. Rebmann, et al., "Large process models: A vision for business process management in the age of generative AI," Künstliche Intelligenz, vol. 39, pp. 81–95, 2025, doi: 10.1007/s13218-024-00863-8.
[14] Z. Shi, J. Dong, and Y. Gan, "Democratizing digital transformation: A multisector study of low-code adoption patterns, limitations, and emerging paradigms," Applied Sciences, vol. 15, no. 12, p. 6481, 2025, doi: 10.3390/app15126481.
[15] A. Rani, D. Mishra, and A. Omerovic, "Enterprise modelling for decision-making in the software ecosystem," Scientific Reports, vol. 15, p. 11134, 2025, doi: 10.1038/s41598-025-90916-1.