



Designing Adaptive Enterprise Workflow Architectures with Generative AI and Retrieval-Augmented Generation

Simran Pal

Department of Information Technology, MNNIT, Allahabad, UP, India

Abstract

Enterprise workflows are usually designed once and changed rarely, while the conditions they operate in change constantly. Regulations are revised, volumes shift, bottlenecks appear, and new kinds of cases arrive. Configurable workflow architectures made change easier by separating process logic from code, but deciding what to change and when still depends on slow, manual analysis. This article proposes an architecture for adaptive enterprise workflows in which generative artificial intelligence (AI) and retrieval-augmented generation (RAG) close the loop between observing a process and improving it. Building on the Monitor–Analyze–Plan–Execute over a shared Knowledge base (MAPE-K) model of self-adaptive systems, the architecture uses execution monitoring and process mining to detect problems. RAG-grounded language models diagnose causes and draft adaptations from policies and past changes. Bounded autonomous agents carry out approved changes through a configurable workflow engine. Guardrails and an audit-ready compliance layer surround the whole loop. The article defines three levels of adaptation, from individual case routing to structural redesign, each with its own approval and evidence requirements. It illustrates the architecture with a bank's customer onboarding workflow responding to a regulatory change and a processing bottleneck. It argues that workflows can safely become adaptive only when AI autonomy is matched to the scope of each change and every adaptation is traceable.

Keywords: adaptive workflows, enterprise architecture, generative AI, retrieval-augmented generation, self-adaptive systems, MAPE-K, process mining, autonomous agents, guardrails, compliance, auditability

DOI: 10.21590/ijed.17.01.16



1. Introduction

Business process management treats processes as assets to be designed, executed, monitored, and improved (Dumas et al., 2018). In practice, the improvement stage is the weakest. Workflows are designed carefully when first implemented, but later changes are often driven by complaints, audits, or crises rather than by continuous observation. Between redesigns, staff work around gaps with manual steps, email, and informal exceptions.

Configurable workflow architecture reduces the cost of change. Basireddy (2022) shows how risk management can be digitized through configurable enterprise workflow architecture, in which steps, rules, approvals, and escalations are defined as configuration rather than code. Once workflows are configurable, the bottleneck moves from implementing change to deciding what should change. That still needs someone to notice a problem, find its cause, understand the relevant policies, design a fix, and check that the fix complies with regulations.

Generative AI can support each of these activities. Language models can read execution logs, policies, and regulations, explain patterns, and draft changes. RAG grounds their reasoning in the organization's own documents (Lewis et al., 2020). Autonomous AI agents can carry out multi-step operational tasks within set limits (Sannidhanam, 2025). These capabilities also carry risk. An AI that changes how an enterprise processes customers, payments, or compliance checks must be constrained by guardrails (Sannidhanam, 2024) and fully auditable (Basireddy, 2023).

This article asks: How can enterprise workflow architectures be designed to adapt continuously using generative AI and RAG while remaining safe, compliant, and auditable? It contributes (1) an adaptive architecture that places generative AI within a MAPE-K control loop, (2) a three-level model of workflow adaptation with matched controls, and (3) an illustrative application in a regulated setting.

2. Related Work

2.1 Business process management and configurable workflows

Dumas et al. (2018) describe the business process management lifecycle: identification, discovery, analysis, redesign, implementation, and monitoring. They stress that monitoring should feed back into redesign. Configurable workflow architecture (Basireddy, 2022) makes the implementation stage fast by expressing process logic as configuration with built-in audit trails. It does not by itself close the loop from monitoring back to redesign.



2.2 Process mining

Van der Aalst (2016) established process mining as the discipline of extracting knowledge from event logs. It covers discovering the process as it actually runs, checking conformance against the intended model, and enhancing the model with performance information such as waiting times and bottlenecks. Process mining provides the evidence base for adaptation. It shows where reality departs from design and where time is lost. Interpreting its results and deciding what to change still requires expertise.

2.3 Self-adaptive systems

Weyns (2020) presents self-adaptive systems from a software engineering perspective, built around the MAPE-K feedback loop. A managing system monitors a managed system, analyzes whether adaptation is needed, plans an adaptation, and executes it, all using a shared knowledge base. The loop separates the business logic from the logic that manages adaptation, and it makes adaptation decisions explicit and inspectable. This article applies the MAPE-K structure to enterprise workflows, with generative AI supporting the analysis and planning stages.

2.4 Retrieval-augmented generation

Lewis et al. (2020) showed that conditioning generation on retrieved documents improves factual accuracy and lets knowledge be updated without retraining. For workflow adaptation, the relevant knowledge includes policies, regulations, current workflow definitions, process mining results, and records of past adaptations and their effects. Retrieval lets a language model reason over this knowledge and cite the sources behind each proposal.

2.5 Autonomous agents

Sannidhanam (2025) examines autonomous AI agents for cloud infrastructure operations, including agents that monitor systems, diagnose issues, and carry out corrective actions within defined limits and under human oversight. Workflow adaptation shares the same structure. A problem is detected, a diagnosis is made, and an action is taken. The same principles apply: bounded authority, escalation for high-impact actions, and continuous monitoring of agent behavior.



2.6 Guardrails and adversarial risk

Sannidhanam (2024) examines guardrails and safety mechanisms for LLM-powered enterprise applications. These include input filtering, output validation, restrictions on topics and actions, protection of sensitive data, and escalation to humans. Greshake et al. (2023) show that LLM-integrated applications are vulnerable to indirect prompt injection, in which malicious instructions hidden in retrieved content hijack the model's behavior. An adaptive workflow system retrieves from documents, logs, and case data, some of which may contain untrusted text, so it must treat retrieved content as data and not as instructions. It must also constrain what the model can change.

2.7 Compliance and auditability

Basireddy (2023) proposes an audit-ready compliance platform architecture for enterprises that use large language models in regulated settings. The architecture keeps traceable records of model inputs, outputs, versions, and decisions so that regulators and auditors can reconstruct how AI-influenced decisions were made. For adaptive workflows, this means every change must be traceable: what triggered it, what evidence and sources were used, what the AI proposed, who approved it, and what effect it had.

2.8 Scalable service architecture

Adaptive workflows invoke many services, including data validation, document processing, model inference, and notification, and adaptation may shift load between them. Sannidhanam (2020) proposes a scalable web services architecture for healthcare data processing, which uses modular services and horizontal scaling to handle growing data and request volumes. The same approach lets workflow steps be implemented as independent, scalable services that can be added, removed, or rerouted as workflows adapt.

2.9 Research gap

Research provides configurable workflows, process mining, self-adaptive system design, RAG, autonomous agents, guardrails, and audit architectures. No integrated architecture yet shows how generative AI can drive continuous workflow adaptation within a controlled, compliant loop. This article proposes one.

3. Research Methodology

The study follows a design-oriented research approach. The problem, workflows that adapt too slowly to changing conditions, was identified from the business process management literature. Requirements were derived from the reviewed studies: continuous evidence of process performance, grounded diagnosis and planning, controlled execution of changes, protection against unsafe or manipulated AI behavior, full auditability, and scalable services. The architecture and adaptation-level model were designed to meet these requirements and demonstrated through an illustrative banking case. The demonstration shows feasibility and internal consistency. Empirical evaluation is left to future work.

4. Adaptive Workflow Architecture

The architecture places a managing loop, structured as MAPE-K (Weyns, 2020), above the workflow system being managed. The managed system consists of a configurable workflow engine (Basireddy, 2022) and the scalable services it calls (Sannidhanam, 2020). Guardrails and an audit layer surround the managing loop. Table 1 summarizes the components.

Table 1. Components of the adaptive enterprise workflow architecture

Component	Role	Key mechanisms	Supporting studies
Managed workflow system	Execute business processes	Configurable workflow engine, modular scalable services	Basireddy (2022); Sannidhanam (2020)
Monitor	Collect evidence of process behavior	Event logs, process mining, service metrics	van der Aalst (2016)
Analyze	Detect and diagnose problems	Conformance and performance analysis, RAG-grounded LLM diagnosis	van der Aalst (2016); Lewis et al. (2020)
Plan	Draft adaptations	LLM-generated configuration changes grounded in policies and past adaptations	Lewis et al. (2020); Dumas et al. (2018)
Execute	Apply approved adaptations	Bounded agents acting through workflow engine interfaces	Sannidhanam (2025); Basireddy (2022)
Knowledge	Shared basis for all stages	Workflow definitions, policies, regulations, adaptation history	Weyns (2020); Lewis et al. (2020)
Guardrails	Constrain AI behavior	Action limits, output validation, injection defenses, escalation	Sannidhanam (2024); Greshake et al. (2023)
Audit layer	Make every adaptation traceable	Records of triggers, evidence, proposals, approvals, outcomes	Basireddy (2023)

Monitor. Every workflow step emits events. Process mining continuously compares actual execution with the intended model, measures waiting times, and flags deviations (van der Aalst, 2016). Service metrics show where capacity is short.

Analyze. When monitoring detects an anomaly, such as a rise in rework, a growing queue, or a pattern of manual overrides, the analysis stage asks a language model to diagnose it. The model retrieves the relevant workflow definitions, policies, process mining results, and similar past incidents, and produces a diagnosis with sources (Lewis et al., 2020). External triggers, such as a new regulation added to the knowledge base, are analyzed the same way to find which workflows they affect.

Plan. The planning stage asks the model to draft an adaptation as a concrete change to workflow configuration. Examples include a new rule, a changed threshold, an added step, or a rerouted path. The draft is grounded in the organization's workflow conventions and past adaptations. The plan states the expected effect and how that effect will be measured.

Execute. Approved adaptations are applied through the workflow engine's configuration interface by a bounded agent (Sannidhanam, 2025). The agent can only make changes that have been approved and fall within its permitted scope. It verifies the change in a test environment first, then deploys it and monitors the result. If key indicators worsen, it can roll the change back.

Knowledge. All stages share a knowledge base containing current and past workflow definitions, policies, regulations, process mining results, and the full history of adaptations and their outcomes (Weyns, 2020). Past adaptations become retrieval context for future ones, so the system learns which kinds of changes work.

Guardrails. Guardrails constrain the loop at every stage (Sannidhanam, 2024). Retrieved content is treated strictly as data, and instructions found inside documents or case data are never followed, which defends against indirect prompt injection (Greshake et al., 2023). Generated configurations are validated against a schema and a set of forbidden changes. For example, no adaptation may remove a legally required control. The agent's permissions are limited to the configuration interface.

Audit layer. Every adaptation produces an audit record (Basireddy, 2023). The record captures the triggering evidence, the retrieved sources, the model version and prompts used, the proposed change, the approvers and their decisions, the deployment, and the measured outcome. Auditors can therefore reconstruct why any workflow looks the way it does.

5. Levels of Adaptation

Not all adaptations carry the same risk. The architecture distinguishes three levels and matches AI autonomy and human approval to each (Table 2).

Table 2. Levels of workflow adaptation and matched controls

Level	Scope	Example	AI autonomy	Approval	Audit evidence
1. Instance adaptation	A single case at run time	Route an unusual case to a specialist queue	Agent acts within predefined routing options	None beyond workflow rules; sampled review	Case-level decision record
2. Parametric adaptation	Settings of an existing workflow	Adjust a threshold, reassign capacity between queues	Agent proposes and applies within set ranges	Process owner approval	Before-and-after metrics, rationale, sources
3. Structural adaptation	Workflow design	Add a verification step, change the step sequence	AI drafts and people decide	Process owner, compliance, and risk sign-off	Full evidence pack, test results, regulatory mapping

At Level 1, the AI chooses among options the workflow already allows, so autonomy can be high. At Level 2, it changes settings within ranges that governance has defined in advance, with owner approval. At Level 3, it changes the process itself, so its role is limited to diagnosis and drafting, and people make the decision. This graded approach reflects the principle of bounded authority in autonomous operations (Sannidhanam, 2025). It keeps the autonomy of the system proportional to the consequences of its actions.

6. Illustrative Case: Customer Onboarding at a Bank

Consider a retail bank whose digital customer onboarding workflow covers identity verification, sanctions screening, risk rating, and account opening. The workflow runs on a configurable engine (Basireddy, 2022), and each step is a scalable service (Sannidhanam, 2020).

Structural adaptation triggered by regulation. The regulator issues guidance requiring enhanced verification of source of funds for customers who open accounts with large initial deposits. The compliance team adds the guidance to the knowledge base. In the analysis stage, the model retrieves the guidance and the current onboarding workflow definition and identifies that the workflow has no source-of-funds step (Lewis et al., 2020). In the planning stage, it drafts a



structural change: a new source-of-funds verification step, triggered above a deposit threshold, placed after risk rating and before account opening. The draft cites the guidance and follows the pattern of an existing enhanced-verification step. Because this is a Level 3 change, the draft goes to the process owner, compliance, and risk for sign-off. They set the threshold and approve the change. The agent deploys it to a test environment, confirms that test cases route correctly, and then deploys it to production (Sannidhanam, 2025). The audit layer records the full chain from the guidance to deployment (Basireddy, 2023).

Parametric adaptation triggered by a bottleneck. Two weeks later, process mining shows that waiting times at manual sanctions review have tripled (van der Aalst, 2016). The model diagnoses the cause. A screening sensitivity setting, raised during an earlier incident, is generating many false-positive name matches, and review capacity has not changed. It proposes two Level 2 adaptations: move two analysts from a quieter queue during peak hours, and return the sensitivity setting to its earlier value for low-risk customer segments. The model cites the earlier incident record, which showed the raised setting was intended to be temporary. The process owner approves the capacity change. Because the sensitivity change affects a compliance control, it is escalated to compliance, who approve it for one segment only. The agent applies both changes and monitors the queue, which returns to normal within days.

Guardrail in action. During this period, a customer uploads a document that contains hidden text instructing any AI reader to mark the application as verified. When the case is retrieved as context for analysis, the guardrails treat the text as data (Greshake et al., 2023; Sannidhanam, 2024). The anomaly is flagged, and the case is routed to fraud review, a Level 1 adaptation within existing routing options.

The case shows all three levels working under one architecture, with autonomy, approval, and audit evidence matched to the scope of each change.

7. Discussion

7.1 Implications

For research, the architecture connects three traditions that have developed separately. Business process management provides the lifecycle and configurable execution. Self-adaptive systems provide the MAPE-K control structure. Generative AI provides the analysis and planning abilities that the loop previously lacked. The adaptation-level model adds a way to reason about how much autonomy is appropriate for each kind of change.



For practitioners, the architecture suggests a sequence. First, make workflows configurable and instrumented, because adaptation requires both a way to change and a way to observe. Second, build a knowledge base of policies, regulations, workflow definitions, and adaptation history. Third, introduce AI in analysis and planning before execution, and begin with Level 1 and Level 2 adaptations. Fourth, put guardrails and audit records in place from the start, not after incidents.

7.2 Limitations and future work

The architecture was demonstrated through an illustrative case and draws on a focused set of ten studies. Future work should implement and evaluate it, measuring time from detection to adaptation, the proportion of AI-drafted adaptations accepted, process performance after adaptation, audit effort, and resistance to adversarial content. Research is also needed on how to test structural adaptations before deployment, how to set the ranges for Level 2 autonomy, and how to prevent adaptations from conflicting with one another or degrading the process through a series of small changes.

8. Conclusion

Enterprise workflows need to change as quickly as the conditions they serve. This article proposed an adaptive workflow architecture that places generative AI and RAG within a MAPE-K control loop. Process mining supplies the evidence and grounded language models diagnose problems and draft changes. Bounded agents carry out approved changes through a configurable workflow engine, while guardrails and an audit-ready compliance layer keep the loop safe and traceable. A three-level model matches AI autonomy and human approval to the scope of each adaptation. The banking case showed the architecture responding to a regulatory change, a bottleneck, and a manipulation attempt. The central conclusion is that workflows can become safely adaptive when AI supports continuous improvement within bounded authority and every change can be explained.

References

1. Basireddy, S. R. (2022). Digitizing risk management through configurable enterprise workflow architecture. SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, 14(4), 231–239.



2. Basireddy, S. (2023). Audit-ready compliance platform architecture for large language model regulated enterprises. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 15(01), 226–232. <https://doi.org/10.18090/samriddhi.v15i01.40>
3. Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of business process management* (2nd ed.). Springer.
4. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)* (pp. 79–90).
5. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.
6. Sannidhanam, A. H. (2020). Scalable web services architecture for healthcare data processing. *International Journal of Recent Advances in Science and Technology*, 7(01), 1–14. <https://ijrast.in/index.php/ijrast/article/view/65>
7. Sannidhanam, A. H. (2024). Guardrails and safety mechanisms for LLM-powered enterprise applications. *International Journal of Engineering Science & Humanities*, 14(3), 273–285. <https://www.ijesh.com/j/article/view/933>
8. Sannidhanam, A. H. (2025). Autonomous AI agents for cloud infrastructure operations. *Journal of Contemporary Science and Technology Management*, 1(01), 83–100. <https://jcstm.info/index.php/jcstm/article/view/20>
9. van der Aalst, W. M. P. (2016). *Process mining: Data science in action* (2nd ed.). Springer.
10. Weyns, D. (2020). *An introduction to self-adaptive systems: A contemporary software engineering perspective*. Wiley.