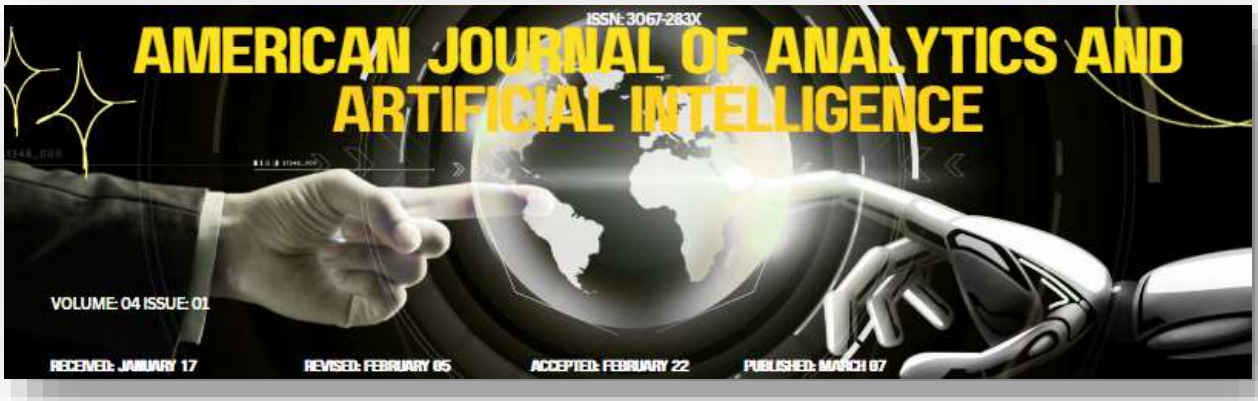


Small Models as Control Planes for Enterprise GenAI

Carlos Mendoza
Asst Professor

Abstract

Generative AI applications such as chatbots and text-to-image systems create demand for GenAI models that address the common information needs of diverse stakeholders in a responsive and personalized manner. Yet the effort to train, host, and serve these models can incur substantial cost and complexity. Many large language models can answer a wide range of questions, but risk being underutilized for specific enterprise workloads. At the same time, enterprise-integrated GenAI services are supporting the digitalization of business processes at an unprecedented scale, revealing latent use cases for specialized models or adjusted configurations of the same model that reflect the cost profiles of these systems. These factors suggest that deploying smaller models to manage the GenAI orchestration layer across an enterprise might yield significant cost savings. A control plane design based on the concept of GenAI orchestration is proposed, along with a set of cost-efficiency principles for implementing this functionality.

Control planes are responsible for decision-making and policy enforcement across a distributed system. Making cost-effectiveness an explicit design goal when architecting an orchestration layer introduces additional considerations beyond those that typically inform the design of control planes. Model size, sparsity, quantization, caching, and workload characterization shape the trade-offs governing the overall cost of model execution, create opportunities for realizing cost savings, and identify workload patterns that can further inform cost-saving measures.

Keywords: Generative AI Orchestration, Enterprise GenAI Control Planes, Cost-Efficient Model Serving, Small Language Models (SLMs), Large Language Model Optimization, Model Sparsity and Quantization, GenAI Workload Characterization, Intelligent Request Routing, Model Caching Strategies, Personalized GenAI Services, Enterprise AI Cost Governance, Control-Plane Decision Logic, Distributed GenAI Architectures, Model Selection and Policy Enforcement, AI-Driven Service Digitalization, Adaptive Model Configuration, Execution Cost Optimization, Latent Enterprise GenAI Use Cases, Scalable GenAI Infrastructure, Next-Generation AI Orchestration Frameworks.

1. Introduction

Generative AI is experiencing a paradigm shift: from creative and entertaining models trained on mass-scale data towards smaller models tailored for business application. Compared to earlier general-purpose models, these customized small models are often cheaper, better aligned to the needs of the target audience, and can use different types of data. However, while serving enterprise needs, they are only partially labeled, not exhaustively governed, and may not be completely correctly functioning. As a result, every instance of using a small model carries risks and potential costs.

Enterprises are likely to put in place controls, governance, and oversight mechan-isms. Costs associated with using small models should naturally also be part of this process. A number of techniques exist to reduce the model cost associated with a workload including caching, workload profiling, model selection, model choice, and sparsity. These techniques can be brought together systematically to provide an integrated cost management layer. To integrate

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 22 PUBLISHED: MARCH 07

these disparate components, consider how enterprise functions typically apply in other areas of software process—typically in a hierarchy of layers or planes. Such layered approach is taken to resource management, and enterprise risk and compliance are treated as an integrated management view for using small GenAI models.

1.1. Overview of the Study

Cost-efficient orchestration of Generative AI-based workflow services—message enrichment, content generation, content moderation, audit trail creation—is vital for digital scaling, yet largely unexplored in enterprise architecture. Sizing, sparsity, and quantization of Small Language Models may drastically reduce cost, albeit at a performance penalty. Semantic similarity provides potential for caching, reusing, and sharing workloads. Orchestration can be abstracted into a layered approach with separate roles for scheduling, messaging, and state management. Within this context, Small Language Models assume critical control-plane roles—making decisions, enforcing policies, allocating resources, and constraining costs.

A cost-efficient approach to integrating Small Language Models as GenAI orchestration control planes within enterprise digital workflows is proposed. First, cost-efficiency determinants are highlighted: model sizing, sparsity, and quantization; caching and workload reusability; workload profiling. Next, a layered orchestration framework is outlined, with dedicated modules for messaging, scheduling, and state management. Finally, the architectural roles of Small Language Models as cost-control mechanisms are presented in further detail. These insights lay the groundwork for empirical validation and explore wider implications for the theory and practice of enterprise architecture.

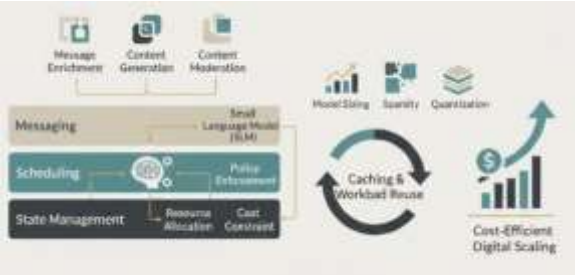


Fig 1: Architecting Cost-Efficient GenAI: Small Language Models as Control-Plane Orchestrators for Enterprise Digital Workflows

2. Conceptual Foundations

Four concepts underpin the proposed approach: control plane, orchestration, GenAI, and enterprise workflow. Control planes manage the flow of data between sub-systems, including the triggering and sequencing of actions. Focusing on GenAI, a sub-set of Artificial Intelligence, addresses enterprise demands for integrated digital workflows around cost and resource utilization—a priority recognized by cloud industry leaders such as Amazon and Google. Enterprises building their own generation models need orchestration layers nearer to the data. Cost-efficient solutions improve control, governance and decision-making by preventing sprawl across the data plane while still benefiting from the data plane's versatility. A delineation between messaging, scheduling, and state management patterns clarifies guarantees and failure handling.

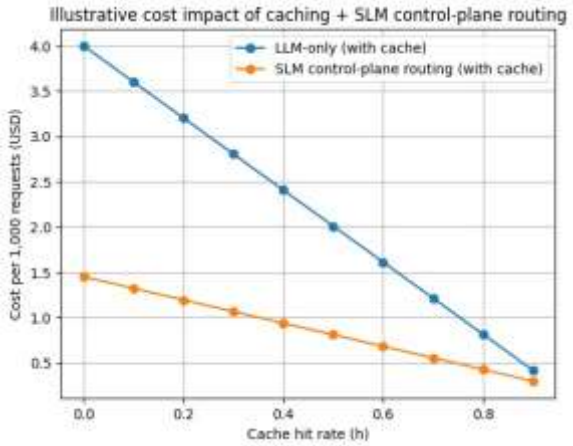
Control planes typically fall into one of three categories: centralized, distributed, or hybrid. Centralized control, a services-oriented approach, uses a single instance that interacts directly with all resources. A distributed approach incorporates control functionality inside the services themselves and delivers requirements to a separate decision-making and scheduling engine. A hybrid approach combines both paradigms.

2.1. Control Plane Paradigms

Control planes may be centralized (one logical instance for all requests), distributed (local instance to handle each

request), or hybrid. Distributed control planes reduce the bottleneck and single point of failure, enabling low-latency decision-making without sacrificing reliability or correctness. Centralized control planes better leverage support knowledge, learning from all users and interactions; they are easier to manage and tune. Proposed architectures combine centralized and decentralized control planes, e.g., as in cloud edge computing. Distributed control is best employed when distributed components have limited knowledge of the system state but require localized and rapid decision-making— feasible with a data-driven model.

For GenAI orchestration, a control plane directly involved in data generation and processing typically incurs higher costs than an interface without processing capabilities. A centralized control plane without access to the data or data-generating services incurs higher latency in data- and metadata-intensive enterprise workflows. Latency-sensitive operations, such as generating manufacturing change orders or operating instructions, impose further requirements: minimized latency, explore-exploit trade-offs, and support for low-cost RL.



Equation 1: Workload and routing variables

Let:

- N = total requests in a window (hour/day/week).
- A control-plane SLM runs per request to enforce policies + routing decisions

Define routing probabilities (workload characterization decides these):

- p_L = probability the request is escalated to a Large Model (LLM).
- $p_S = 1 - p_L$ = probability the request is handled by a Small Model (SLM).

2.2. GenAI Orchestration in Enterprise Contexts

Enterprise-managed digital workflows integrate a variety of services into a coherent process responding to business needs. Although most services offer an API for interaction, managing non-obvious dependencies, data-sharing contracts, state-keeping, sequencing, and all related orchestration skills remains a daunting task involving a human operator or support team well-versed in the process details. Teenage users of social media may have figured out the recipe for making a specific TikTok video go viral, but that process typically does not scale well amid the stream of day-to-day DM traffic because there remain scale and personnel bandwidth limits.

Enterprises increasingly integrate generative AI services into their digital workflows and explore the potential for orchestrating these services as part of a coherent workflow. Existing digital workflows and orchestration processes have tended to come into being ad hoc or as coipied recipes; however, a more modular enterprise-ready approach is emerging to treat each part as a reusable component. Such a growing reusable orchestration layer acts as an enterprise control plane actively handling all of the orchestration required to connect independent service modules into a coherent process.

Cache hit rate (h)	LLM-only cost / request ($)	SLM control-plane cost / request ($)	Savings vs LLM-only (%)
0.0	0.004	0.00145	63.74999999999999
0.1	0.003602	0.001322	63.298167684619656
0.2	0.0032040000000000003	0.001194	62.73408239700375
0.3	0.002806	0.001065999999999999	62.00997861724875
0.4	0.002408	0.000937999999999999	61.04651162790697

3. Architectural Roles of Small Language Models

Within layered orchestration frameworks, small language models (SLMs) can assume two distinct roles that are traditionally fulfilled by separate systems. First, SLMs can make decisions as a control plane, rendering them responsible for governance, risk mitigation, compliance, and related topics. Used in this manner, SLMs act similarly to a metaplane that operates in parallel to the data plane or an associated management plane; they engage at critical decision points throughout the process and are responsible for enforcing constraints that satisfy enterprise-level requirements such as data governance, privacy, compliance, quality, and risk mitigation. Key considerations when designing cost-efficient SLMs for these purposes include sizing, sparsity, and quantization. Second, SLMs can function as an economic control plane whose objectives span resource allocation, consumption management, and cost containment. By controlling the usage patterns of larger models specifically to fit within an enterprise-level budget, a small instance of a generative model can be applied constantly without overshooting or undershooting the target consumptive profile. Considerations in this domain include automating the adjustment of model sizes based on predicted workload, using an internal cache to reduce the amount of externally requested capacity, and anticipating workload characterizations early in the process. Authorship of digital content produced by generative models is a critical issue that combines aspects of both roles and, when managed carefully, allows easy fulfillment of IAM requirements and, indeed, the adoption of IAM technology and concepts into the GenAI orchestration strategy.

3.1. Decision-Making and Policy Enforcement

Enterprise Digital Workflows demand resource allocation decisions and policy enforcement with respect to compliance, risk mitigation, and governance requirements. Cost control through model usage governance and permission allocation along with autoscaling and SLA adherence Mechanisms ensure that small language models can orchestrate enterprise GenAI operations cost-efficiently.

Governance guarantees provide a reasonable assurance that model reuse is constrained to maintain data and model quality and provide high-quality response output. The governance guarantees specify rules around acceptable data provenance, quality thresholds, and audit checks. Risk mitigation guarantees prevent the generation of unacceptable content. Compliance guarantees support meeting pre-defined standards or legislation when addressing a user request. Permissions indicate whether a request can be satisfied by allowing the use of a language model. If permission is granted, the model address is updated and the re-evaluated usage limits enforced. Autoscoping and SLA management allocate computing capacity for dataspace workloads, adjusting it to fit the service-level objective. As an additional cost-control mechanism, the maximum number of LPM instances that can concurrently handle LPM messages is also adjusted with the help of an associated auto-scaler.

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 22 PUBLISHED: MARCH 07

Fig 2: Governed Orchestration: A Framework for Cost-Efficient Policy Enforcement and Risk Mitigation in Enterprise GenAI Workflows

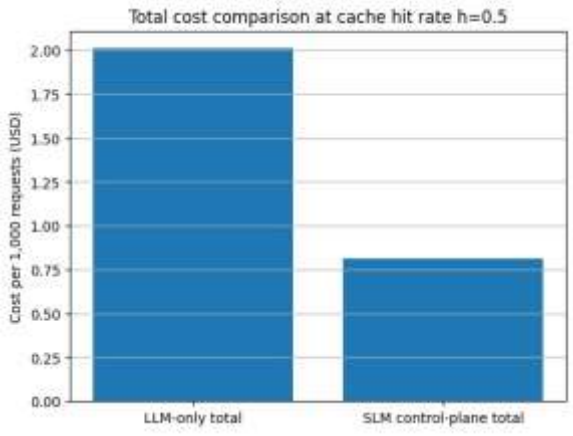
3.2. Resource Allocation and Cost Control

Action control planes make wise decisions regarding resource allocation, model usage, and cost control in Layered Orchestration Frameworks. The associated control tasks monitor the execution of the data-processing operations making up the orchestration, using feedback loops to validate that the results meet quality guarantees. When operations appear outside the expected boundaries, they can be remediated by adjusting message parameters or selecting different resources, service templates, or execution versions. Indeed, message parameters often govern information-quality dimensions, such as the precision of a forecast. The control tasks continually evaluate these budgets and constraints, adjusting resource allocation and scaling in accordance with the service-level agreements (SLAs) given by other digital-layer actors. Action control planes adjust the allocation and selection of models whenever breaks in contract observance occur. Complementary to this operation, they also take care of probabilistic budget limits, which provide overuse indicators for one or several engines contributing to a service–level agreement. When budgets reach or exceed their limit in a session, action control planes consolidate the usage information of this session's path across the cost control plane. Since costs are often associated with Latency, implementation heuristics support the correct adjustment of caching and reuse mechanisms used by resources being observed.

4. Layered Orchestration Frameworks

Several dimensions can be abstracted out of an orchestration workflow and exposed through clearly defined interfaces. These interfaces not only separate an orchestration workflow into an ensemble of cooperating, communicating sub-workflows but also provide the means for defining the choreography of those interactions. Messaging, scheduling, and state management are three critical dimensions of any orchestration architecture. Established distributed-computing principles point to a continuum of architectures for the implementation of each dimension, each with its advantages and drawbacks. Applying these principles should reveal the appropriate shape of orchestration for any particular use case. Messaging can be implemented in a point-to-point fashion, where workflows explicitly send messages to one another, or through a message broker, where any entity can publish messages and where interested entities can subscribe to messages of interest. Point-to-point messaging is often easier to reason about, but is also more brittle. With a broker, messages are loosely coupled as senders do not need to know who the recipients are, and additional recipients can be added without impacting existing logic. However, messages can be harder to trace or control. Using a broker allows decoupled communication, discovery, and scaling of producers and consumers, often with greater resilience. Patterns such as request-reply, publish-subscribe, and scatter-gather become easier to implement. At the same time, coordination might still need to be orchestrated rather than left to the natural flow of data.

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 22 PUBLISHED: MARCH 07



Equation 2: Token-based inference cost (most common in GenAI billing)

For a model m, define:

- t_{in}, t_{out} = input/output tokens for a request,
- $\pi_{in}^{(m)}, \pi_{out}^{(m)}$ = price per input/output token.

Per-request model cost:

$$C_m = t_{in}\pi_{in}^{(m)} + t_{out}\pi_{out}^{(m)}$$

If you assume average tokens $\bar{t}_{in}, \bar{t}_{out}$, then the **expected per-request**:

$$\mathbb{E}[C_m] = \bar{t}_{in}\pi_{in}^{(m)} + \bar{t}_{out}\pi_{out}^{(m)}$$

4.1. Abstraction Layers and Interfaces

Abstraction layers and interfaces are required to modularize enterprise-integrated GenAI orchestration into manageable components. Each layer requires a clear contract describing what its modules exchange and the roles their modules fulfill. These contracts streamline implementation, make connecting layers easier, and clarify the relationships among modules that occur at the same layer.

The modular boundaries, data models exchanged between adjacent layers, and responsibilities of each layer are determined by the nature of the digital workloads being orchestrated. A workflow-centric perspective that divides orchestration decisions into three categories—messaging, scheduling, and state management—provides the highest-level specification. Respective patterns, latency, throughput, consistency, and fault-tolerance guarantees—and therefore failure paths—are then identified. These characterize the layers without confining their physical architecture.

4.2. Messaging, Scheduling, and State Management

Responsibility for messaging, scheduling, and state management among the composition layers is now examined. Messaging is a fundamental enterprise workflow operation, hence accompanying requirements and guarantees warrant explicit articulation. Control-plane data exchanges should be responsive, aligned with end-user needs and service-level agreements (SLAs) on latency. Scheduling concerns are application specific; guaranteeing a certain throughput per workflow class may be important. Beyond batch processing, state management is central to the distributed orchestration paradigm: structure, data integrity, and consistency of all shared resources forged during workflow processing must be addressed. Transactional models, operational transformation, calculus-based, and other solutions are suitable if their guarantees are met within workload characteristics.

Provenance and data-governance guarantees represent a particular category of guarantee, substantiating the trustworthiness of processing outcomes. Interest in verified AI intensifies with the deployment of foundation models as scientific advisors. Superior AI-generated models may be invalidated by untrusted input data stemming from ungoverned processes. Provenance definitions capture all aspects related to the input data lineage and can be extended to such intrinsic properties as quantification, generated by the metadata space of a virtual ontology. Combining the operations of forecast, classification, and categorization achieves great efficiency, yet enriching the content with forwarded metadata is crucial.

5. Design Principles for Cost-Efficiency

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 22 PUBLISHED: MARCH 07

Enterprise-integrated generative AI orchestration entails considerable costs in terms of model execution as well as external resource consumption. Cost control requirements can therefore shape design decisions. In absence of a well-defined SLA and commitment to a formal budget, SPMLM execution overhead risks constraining the financial feasibility of the overall workflow. Suitable principles should hence drive model sizing choices and the spatial, temporal, and functional characteristics of model activation.

The enterprise context also opens up possibilities for SPMLM execution costs to be reduced through adaptive approaches. For example, resource allocation decisions achieved through proper management-at-runtime techniques can ensure that indeed only the necessary model sizes are executed without leading to a violation of budget allocation. Caching policies can reduce the need for re-executing an SPMLM task generating the same output for the same or similar input conditions (weight means that the inputs vary not so much that their meanings differ). Workload characterisation supports the understanding of the workload constraint and the adaptation of the architecture or the implementation accordingly.

5.1. Model Sizing, Sparsity, and Quantization

Cost-efficiency is paramount given the rapid increase in the cost of deploying large GenAI models and the expected continued increase in their usage. Sizing models properly is therefore critical and efforts should be made to ensure large models only run when required. Sparsity of both the models and the requests can help bring down costs and quantization can often provide good speedup with minimal cost in terms of generation quality. It should be noted that such measures come at the cost of accuracy and performance. Caching, reuse and request profiling are therefore instrumental in controlling costs while achieving good latency performance by avoiding indiscriminate caching – adopting cache eviction policies matching the actual usage patterns – and ensuring the reuse patterns of previous workload segments can be learned and exploited during execution.

Model Sparsity at Inference Time is often rarely addressed in deployment considerations but can be exploited if the model and its use case permit it. Prompts containing code with the associated problem/component being within acceptable similarity bounds can exploit such sparsity. The selection of a model capable of executing the task on limited compute or memory resources greatly speeds up the generation and allows for lower cost GPU services. A similar approach can be attempted for prompt similarity with respect to locality. When certain prompts recur often enough, the first few request responses can be cached followed by an LRU cache eviction strategy until the recency of the requests–response pairs can be predicted in order to switch to a predictive cache eviction algorithm. Request profiling can further assist in exploiting the identified reuse or caching patterns.

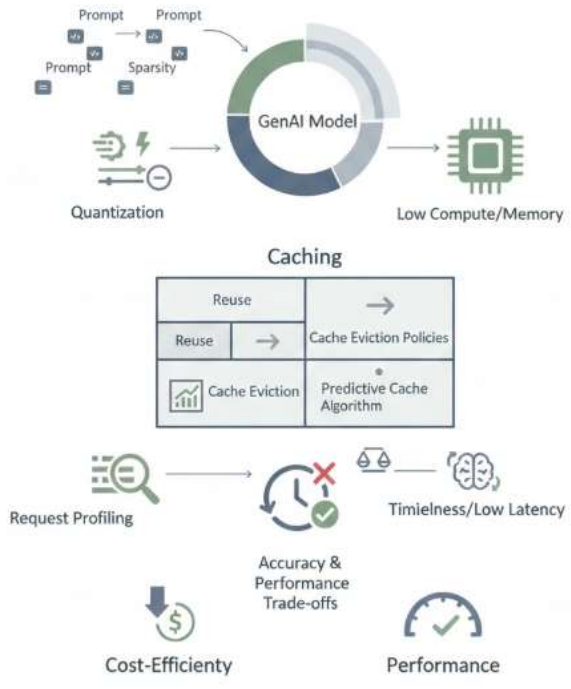


Fig 3: Optimizing Generative AI Deployment: A Framework for Cost-Efficient Inference through

Sparsity, Quantization, and Predictive Cache Orchestration

5.2. Caching, Reuse, and Workload Characterization

Caching, reusing, and profiting from knowledge about resource-intensive tasks and results reduces the overall cost associated with service invocation. High resource consumption in a GenAI service run can be a barrier for generating low-value content, e.g. fault-tolerant infrastructure assets, noisy-source network blueprints. For such services, generating the response once and caching the result can lead to overall lower cost. Patterns of reuse can be identified based on application domain. For instance, translations of content (blog posts and captions) into multiple languages tend to be invoked at least twice. Interest in workload profiling has increased with the advent of Generative AI and increased shifts from scripted code flows to simpler Liquid-like mechanisms in Digital Process Automation spaces. Exploring but (or by) pinpointing where workloads naturally flop and heat need specialization remain hot topics across Cloud and Datacenter designs – also DGX-Farm-design studies. Benefit of workload profiling is magnified in Inference workloads when inferencing either over a very small model or specialized model being orders of magnitude recharge scale/duration lower then the model being used or depot infrastructure being orders of magnitude slower than any model in use.

6. Integration with Enterprise Digital Workflows

Orchestration layers that exploit small language models as control planes must abide by enterprise workflow requirements for digital data acquisition, processing, and synthesis. Three enterprise data pillars govern rigorous digital workflows: provenance, identity and access management (IAM), and data quality.

Provenance ensures the source of all data and service output is known. The customer must know where or whom data originated from, how it was generated, how the digital service performed the work, and how it was fed to the next stage in the digital workflow. Without tracking, external data can be misleading or factually incorrect; digitally, this can create a chimera effect, where different outputs cannot be reconciled. Provenance tracking also encompasses data quality and auditability. At every workflow execution step, the generated output must adhere to the expected data quality surface. For every traditional process, quality checks ensure predefined career actions are carried out. For data-oriented services, this is done through tracking data shape—unexpected shape for an image-processing service output—or unexpected model confidence scores.

IAM ensures every actor, both human and digital, is correctly authorized to perform specific actions within their defined roles. IAM systems leverage four pillars to guarantee that digital services or users cannot violate segregation of duties: authentication, approval, authorization, and audit logging. Enterprise workflow applications cannot forget the IAM implantation happening across every digital service being orchestrated. Control planes should therefore invoke the IAM policies of every service at every stage.

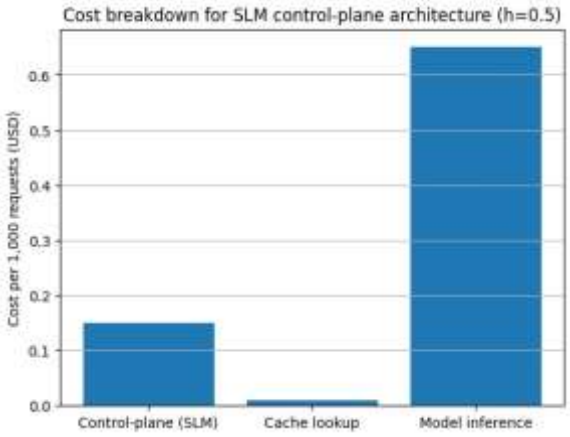
6.1. Data Governance and Provenance

Enterprise digital workflows generate diverse types of data in support of different processes, function domains, and business units; as aligned with the organization's objectives, vision, and mission. These data generated are not only voluminous but also critical to the business objectives, driving cost structures. They must, therefore, be governed for use in enterprise GenAI use cases, requirements, and workloads.

Governance, especially data lineage, quality, auditability, and a single source of truth, must be built into these GenAI use cases, solutions, and workloads. Quality control must also be implemented through structured prompts and API standardization. Regularized requirements for working with GenAI afford opportunities to explore data for quality, gaps, and other necessary enhancements. The investment to satisfy future working GenAI workloads can thus be amortized over the required frequency while also raising

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 22 PUBLISHED: MARCH 07

the quality, coherence, and result fidelity of the enterprise GenAI outputs.



Equation 3: Caching/reuse equation (semantic similarity reuse)

Let:

- h = cache hit rate
- C_{lookup} = cost of cache lookup (embedding + vector search + retrieval overhead).

Then **expected per-request cost for "model inference with cache"** for a given model m:

$$\mathbb{E}\left[C_{m,\text{with cache}}\right] = h \cdot C_{lookup} + (1-h) \cdot C_m$$

6.2. Identity, Access, and Authorization

Enterprise Digital Workflows demand strict identity and access management across all services, which require careful integration with existing Identity and Access Management (IAM) services. Each service must authenticate invoking identities and ensure either ownership or authorized permissions over the resources being accessed. The IAM system must manage role memberships and apply services to the invocation context. IAM gathers and enforces fine-grained service policies such as those controlling service invocation, method access rights, input data access and data transformation services. Since the orchestration control plane interacts with many services, an explicit authorization check for every interaction would lead to performance overhead. For that reason, IAM should streamline these checks by tagging policies with relevant role memberships. An IAM service for Digital Workflows should provide a single point for identity and access management, including role definition and association, permissions over invoked services and resources, policy creation and adaptation for invoking services, automatic tag generation for orchestration service layers, and dependencies over data transformation services. Considerations for making models comply with involved authorizations policies can further appear within the Decision Making role or under Policy Control. Decisions issuing calls to other services that will invoke yet another service should ensure these are not redundant and permitted by applied IAM policies, especially in distributed control planes. Such policies can be central, ensuring authorizations over defined roles and memberships, or simply and adequately stated by every model at use.

7. Conclusion

Models that are significantly smaller than the workhorse paragon can be valuable adjuncts. Evidence suggests that they can be employed in control-plane roles that permit power and performance consuming model use to be effectively managed through concept- and workload-specific autoscaling and batching strategies. Doing so can simultaneously satisfy the Latency, Throughput, and Cost guarantees expected of service-level objectives. Centralized decision-making without clear resourcing guarantees raises questions of model overload, bottlenecks, and lower-layer inefficiencies.

By making the use of power and performance consuming models a capability, rather than a requirement, their deployment enters new operating regimes. Resource constraints make their retention unusable and temporarily uncacheable responses to data requests that are not budgeted and granted subsequently can be reused when revisited against suitably profiled workloads. The analysis of GeneticAI application calls for empirical study and the development of theory-backed patterns, best-practice

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 05 ACCEPTED: FEBRUARY 22 PUBLISHED: MARCH 07

solution maps, and common standards and APIs. The enterprise domain naturally emerges as a first testbed that is sensitive to costs and resource contention, holds its OASIS industry association, and manages business process integration smartly with DMN and BPMN standards.

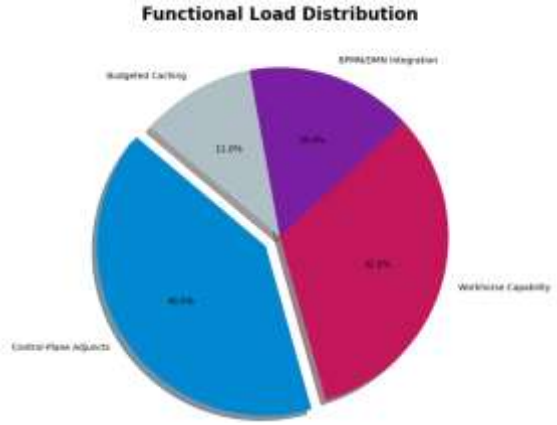


Fig 4: Functional Load Distribution

7.1. Future Directions and Implications for Research

Model size controls inference cost. Small Language Models (SLMs) reduce latency and improve throughput by decreasing the inference time per call. Large Models (LMs) outperform SLMs and specialized models in many benchmarks, provided the physical resources are available. As training cost is only partially mitigated by a little-exploited reuse during inference, tailored solutions for larger workloads are often superior. Sparsity, parallel-quantization, distillation, and mixture-of-experts strategies are other approaches to trim costs with respect to different target metrics depending on the architecture. Caching latencies, results, or both can help balance the cost of call overhead or poor reuse. Popular word-predicting patterns of GenAI requests can be exploited for workload profiling, allowing to anticipate work peaks, cache available resources, or adjust the spendings of infrequent auto-scaling.

The command-and-control approach of GenAI shortcuts the specification of edge-processing logic and stands at a plane-level below the management-scaling proposal of self-driving cloud optimizers. Empirical studies on the design and control of cost-efficient Distributed-Multi Commonwealth GenAI Orchestration Layers grounded on GenAI verticals are an urgent task toward operational-ready enterprise-integrated Digital Workflows. This integration accelerates their path toward maturity and enables an emergence of open standard guidelines to alleviate the adoption of GenAI solutions.

8. References

1. Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, 23(120), 1–39.
2. Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 3045–3059.
3. Ganesh, S. K., Subbareddy, K., Gopi, A., Davuluri, P. N., Mannar, B. R., & Karnawat, A. T. (2026, June). Fraudulent Credit Card Transaction Detection Using a Hybrid Ensemble-Anomaly Detection Framework. In 2026 6th International Conference on Intelligent Technologies (CONIT) (pp. 1-6). IEEE.
4. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. International Conference on Learning Representations.
5. Li, Y., Wei, F., Zhang, J., & Zhang, X. (2023). ELLA-V: Stable neural network adaptation for efficient language model serving. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 1–15.
6. Yandamuri, U. S., Loganathan, R., Davuluri, P. S. L. N., Rani, P. R. S., Kolla, S. H., & Nagubandi, A. R. (2026). Adaptive Intelligence Networks for Humancentered Enterprise Automation and Governance. In 2026 Third International Conference on Innovations in Cybersecurity and Data Science (ICICDS) (pp. 678–683). IEEE. 2026 Third International Conference on Innovations in

Cybersecurity and Data Science (ICICDS). https://doi.org/10.1109/icicds70526.2026.11604640

7. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. Advances in Neural Information Processing Systems, 36, 10088–10115.
8. Chen, L., Zaharia, M., & Zou, J. (2023). FrugalGPT: How to use large language models while reducing cost and improving performance. arXiv preprint arXiv:2305.05176.
9. Banoth, S., Santoshi Kumari, M., Lalitha, P., Deepthi, P., Kumar Peddi, R., & Kavitha, P. (2026). An Efficient Hybrid K-Means and Random Forest-Based Approach for Cloud Malware Detection and Privacy Protection. In 2026 6th International Conference on Intelligent Technologies (CONIT) (pp. 1–6). IEEE. 2026 6th International Conference on Intelligent Technologies (CONIT). https://doi.org/10.1109/conit69683.2026.11621822
10. Jiang, D., Ren, X., & Lin, B. Y. (2023). LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 14165–14178.
11. Mangalampalli, B. M., Peddi, R. K., Kolla, S. K., Reddy, V. A. R., Mangala, N., & Seenu, A. (2026, June). Explainable Clinical Graph Intelligence Framework for Longitudinal Risk Modeling and Care Pathway Optimization. In 2026 Third International Conference on Innovations in Cybersecurity and Data Science (ICICDS) (pp. 1-6). IEEE.
12. Leviathan, Y., Kalman, M., & Matias, Y. (2023). Fast inference from transformers via speculative decoding. Proceedings of the 40th International Conference on Machine Learning, 19274–19286.
13. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., & Stoica, I. (2023). Efficient memory management for large language model serving with PagedAttention. Proceedings of the 29th Symposium on Operating Systems Principles, 611–626.
14. Mahadevan, S. (2026). Governed Serverless Automation Ecosystem for AI-Driven Enterprise Integration and Sustainable Cloud Operations. International Journal of Computer Information Systems and Industrial Management Applications, 18(16s), 1561-1570.
15. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36, 68539–68551.
16. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
17. Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., et al. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.
18. Vakkalagadda, T., & Rajamanickam, V. (2026). Agentic AI Architectures for Next-Generation Investment Advisory and Portfolio Intelligence. International Journal of Computer Information Systems and Industrial Management Applications, 18(9s), 1206-1219.
19. Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Ruehle, V., Lakshmanan, L. V. S., & Awadallah, A. (2024). Hybrid LLM: Cost-efficient and quality-aware query routing. International Conference on Learning Representations.
20. Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., et al. (2024). Mixtral of experts. arXiv preprint arXiv:2401.04088.
21. Abdin, M., Jacobs, S. A., Awan, A. A., Aneja, J., Awadalla, H., Awadallah, A., et al. (2024). Phi-3 technical report: A highly capable language model locally on your phone. arXiv preprint arXiv:2404.14219.
22. Bhavani, B. D., SR, S., Loganathan, R., & Nagaraj, S. (2026, April). Evolutionary Gravitational Neocognitron Neural Network, Snow Leopard Optimization and Deep Graph Reinforcement

Learning for Routing Protocol in WSN. In 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN) (pp. 1-8). IEEE.

23. Mei, K., Zhu, X., Xu, W., Hua, W., Jin, M., Li, Z., Xu, S., Ye, R., Ge, Y., & Zhang, Y. (2024). AIOS: LLM agent operating system. arXiv preprint arXiv:2403.16971.
24. Zhuang, N., Tao, M., Zhang, C., Jin, Y., Xu, K., Chen, L., Huang, S., & Feng, Y. (2024). Harder task needs more experts: Dynamic routing in MoE models. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 12883–12895.
25. Piyush Kumar Pareek. (2026). Human-Centric Machine Learning Frameworks for Scalable Software Quality Prediction. Journal of Intelligent Decision Making and Information Science, 3(5s), 1525–1543. https://doi.org/10.59543/jidmis.v3.1284
26. Araujo, V., Moens, M.-F., & Tuytelaars, T. (2024). Learning to route for dynamic adapter composition in continual learning with language models. Findings of the Association for Computational Linguistics: EMNLP 2024, 687–696.
27. Ge, Y., Ren, Y., Hua, W., Xu, S., Tan, J., & Zhang, Y. (2023). LLM as OS, agents as apps: Envisioning AIOS, agents and the AIOS-agent ecosystem. arXiv preprint arXiv:2312.03815.
28. Loganathan, R. (2026). SaaS Entitlement Governance: A Reproducible Model for Detecting and Reclaiming Over-Provisioned Access at Enterprise Scale. Journal of Intelligent Decision Making and Information Science, 3(7s), 2519-2530.
29. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. International Conference on Learning Representations.
30. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36, 68539–68551.
31. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, J., et al. (2024). A survey on large language model based autonomous agents. Frontiers of Computer Science, 18, 186345.
32. Davuluri, P. S. L. (2023). AI-Augmented Sanctions Screening: Enhancing Accuracy and Latency in Real Time Compliance Systems. AI-Augmented Sanctions Screening: Enhancing Accuracy and Latency in Real Time Compliance Systems (December 15, 2023).
33. Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., et al. (2023). The rise and potential of large language model based agents: A survey. arXiv preprint arXiv:2309.07864.
34. Wang, Y., Zeng, Z., Li, Y., Huang, Z., Chen, Z., & others. (2024). A survey on large language model based autonomous agents. ACM Computing Surveys, 57(6), 1–45.
35. Behera, A. P., Champati, J. P., Morabito, R., Tarkoma, S., & Gross, J. (2025). Towards efficient multi-LLM inference: Characterization and analysis of LLM routing and hierarchical techniques. arXiv preprint arXiv:2506.06579.
36. She, J., Zheng, W., Liu, Z., Wang, H., Xing, E. P., Yao, H., & Ho, Q. (2025). Token level routing inference system for edge devices. Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 1–10.
37. Ahuja, K., Kumar, S., Goyal, N., & others. (2025). Adaptive LLM routing under budget constraints. Findings of the Association for Computational Linguistics: EMNLP 2025, 11320–11340.
38. Zhang, D., Song, J., Bi, Z., Song, X., Yuan, Y., Wang, T., Yeong, J., & Hao, J. (2025). Mixture of experts in large language models. arXiv preprint arXiv:2507.11181.
39. Johnson, W., & Lee, C. (2026). Evaluating small language models for front-door routing: A harmonized benchmark and synthetic-traffic experiment. arXiv preprint arXiv:2604.02367.
40. Piyush Kumar Pareek. (2026). Self-Evolving Analytics Pipelines for Reliable AI-Augmented Software Systems. Journal of Intelligent Decision

Making and Information Science, 3(5s), 1558–1578. https://doi.org/10.59543/jidmis.v3.1286

41. Li, J. (2026). The evolution of mixture-of-experts architectures in large language models: Routing, topology, load balancing, and expert parallelism. arXiv preprint arXiv:2608.08650.