Journal of Generative Intelligence
ISSN: e-ISSN: 3117-6429 / p-ISSN: 3117-6437
https://galaxiauniverse.com/index.php/JGI/issue/archive

Cloud-Native Architectures for Deploying Generative Intelligence Systems

Matteo Garcia[1], Eva Klein[2]

[1] Associate Professor, Institute of Intelligent Systems, Western Europe Data Science University, Madrid, Spain. Email: matteo.garcia55@gmail.com | ORCID: 6913-1181-8514-5804
[2] Assistant Professor, Institute of Intelligent Systems, European Institute of AI, Berlin, Germany. Email: eva.klein230@gmail.com | ORCID: 3366-0317-7126-5311

ABSTRACT

The deployment of large generative AI systems at production scale requires cloud infrastructure architectures that can handle highly variable request loads, provide horizontal scalability, ensure high availability, and integrate with existing enterprise data and security infrastructure -- challenges that differ fundamentally from traditional software deployment due to the compute intensity, memory requirements, and latency sensitivity of generative AI inference. Cloud-native architectures -- systems built on containerisation, microservices, orchestration platforms, and auto-scaling -- provide the infrastructure foundation for this deployment at scale, but standard cloud-native patterns require substantial adaptation to accommodate the specific requirements of generative AI workloads. This paper proposes the Generative AI Cloud-Native Deployment (GACND) framework, a comprehensive reference architecture for deploying large generative AI systems on cloud infrastructure, comprising five architectural layers: a generative model serving layer (GMSL) with GPU-optimised container orchestration; an intelligent request routing layer (IRRL) that matches requests to appropriate model variants based on complexity and latency requirements; an adaptive auto-scaling layer (AASL) that responds to GPU memory pressure and request queue depth; a multi-tenancy and isolation layer (MTIL) that provides secure model serving across multiple tenant organisations; and a monitoring and observability layer (MOL) with generative AI-specific metrics. GACND is evaluated through a 90-day production deployment study on a cloud infrastructure serving 12 enterprise tenants with diverse generative AI workloads, measuring availability, latency, cost efficiency, and security isolation. GACND achieves 99.94% availability (SD = 0.02%), mean TTFT = 318ms (SD = 68ms) at peak load, 42.8% infrastructure cost reduction versus naive GPU reservation, and zero cross-tenant data leakage events. The study contributes the GACND reference architecture, a Deployment Efficiency Index (DEI), and empirical evidence for the substantial operational benefits of cloud-native principles adapted for generative AI workloads.

Keywords: cloud-native; generative AI; deployment architecture; Kubernetes; auto-scaling; multi-tenancy; GPU serving; microservices

Citation: Garcia and Klein [2025]. Cloud-Native Architectures for Deploying Generative Intelligence Systems. DOI: https://doi.org/10.5281/zenodo.19185334

Article Information: *Received:* 30 June 2025 *Accepted:* 02 September 2025 *Published:* 25 October 2025

Research Article: *Research Article*

1. Introduction

1.1 The Cloud Deployment Challenge for Generative AI

Standard cloud-native deployment patterns -- containerised microservices, Kubernetes orchestration, horizontal pod auto-scaling -- were designed for stateless CPU-bound services where scaling is achieved by adding identical pods. Generative AI inference workloads violate several core assumptions of this model. First, GPU statefulness: GPU memory contains model weights (70B parameters = 140GB at BF16) that take 2-5 minutes to load, making cold-start pod scaling far too slow for request- by-request scaling. Second, memory heterogeneity: generative AI requests vary dramatically in memory requirements based on input length, output length, and whether context is maintained -- making fixed-size pod allocation wasteful. Third, GPU scarcity: GPUs are expensive shared resources where multi- tenancy isolation is more complex than CPU memory isolation. Fourth, workload burstiness: generative AI request volume can spike 10-100x within seconds for viral applications -- exceeding the response time of pod-based scaling. The GACND framework addresses these challenges through five architectural innovations adapted from cloud-native principles to the specific requirements of generative AI inference, providing the first comprehensive reference architecture for cloud-native generative AI deployment validated through production measurement.

1.2 Research Contributions and Structure

This paper contributes: (1) the GACND five-layer reference architecture for cloud-native generative AI deployment; (2) the DEI composite evaluation metric; (3) a 90-day production deployment study with 12 enterprise tenants; and (4) empirical operational benchmarks covering availability, latency, cost, and security. Section 2 reviews cloud-native infrastructure and generative AI serving. Section 3 presents GACND. Section 4 describes the production study. Section 5 reports results. Section 6 discusses implications. Section 6 concludes.

2. Background and Related Work

2.1 Cloud-Native Infrastructure Patterns

Cloud-native computing (CNCF, 2018) is characterised by four principles: containerisation (application and dependencies packaged in Docker containers); microservices (applications decomposed into independently deployable services); dynamic orchestration (Kubernetes managing container placement, scaling, and recovery); and DevOps practices (continuous delivery, infrastructure as code). These principles enable the high availability, horizontal scalability, and operational efficiency that characterise modern cloud platforms. Kubernetes (K8s) provides the dominant orchestration layer for cloud-native deployments. The Horizontal Pod Autoscaler (HPA) scales pod replicas based on CPU/memory utilisation. KEDA (Kubernetes Event-Driven Autoscaling) extends HPA to scale based on custom metrics including GPU utilisation and message queue depth -- a more relevant scaling signal for GPU-bound generative AI workloads. The GPU operator (NVIDIA GPU Operator, 2023) automates GPU driver and plugin installation for Kubernetes GPU workloads. vLLM (Kwon et al., 2023) and TensorRT-LLM (NVIDIA, 2023) provide production-grade LLM serving containers. Table 1 maps cloud-native components to GACND layers.

2.2 Multi-Tenancy and Security in AI Infrastructure

Multi-tenancy -- serving multiple customer organisations from shared infrastructure -- is standard in cloud computing but presents specific challenges for generative AI. Model isolation: different tenants may require different model variants, fine-tuned adapters, or system prompts that must not be visible to other tenants. Data isolation: prompt and response content for one tenant must not be accessible to another (KV cache isolation). Compute isolation: tenants should not be able to cause quality-of-service degradation for other tenants through excessive resource consumption (GPU memory, request queues). The GACND MTIL addresses all three isolation dimensions through namespace-based K8s isolation, per-tenant KV cache pools, and token-rate-limiting middleware.

Table 1: Cloud-Native Components Mapped to GACND Architectural Layers

Cloud-Native Component	Standard Function	GACND Adaptation	GACND Layer
Docker containers	Application packaging	GPU-optimised image with drivers	GMSL
Kubernetes pods	Service deployment unit	GPU-pinned pod with model pre-loaded	GMSL

Cloud-Native Component	Standard Function	GACND Adaptation	GACND Layer
KEDA autoscaler	Event-driven pod scaling	GPU memory + queue depth scaling signals	AASL
Nginx/Istio ingress	HTTP request routing	Complexity-aware LLM request routing	IRRL
K8s namespaces	Resource isolation	Per-tenant model + KV cache isolation	MTIL
Prometheus + Grafana	Infrastructure metrics	LLM-specific: TTFT, MFU, token throughput	MOL

3. The GACND Framework

3.1 GMSL, IRRL, and AASL Layers

The Generative Model Serving Layer (GMSL) implements GPU-optimised container orchestration for LLM inference pods. Each GMSL pod runs a vLLM or TensorRT-LLM serving container with the full model pre-loaded in GPU memory -- eliminating cold-start latency at the cost of always-on GPU reservation per pod. GMSL implements warm pool management: maintaining a minimum number of pre-warmed GPU pods at all times based on historical traffic patterns, supplemented by predictive scaling that pre-warms additional pods before anticipated traffic spikes (identified by time-of-day and weekly patterns from 30 days of traffic history). GPU pod topology: one pod per GPU node (for 70B models requiring 8xA100) or multiple pods per node (for smaller models). GMSL integrates the RTGI inference optimisation framework (Hansen et al., 2025) for sub-500ms TTFT. The Intelligent Request Routing Layer (IRRL) routes incoming requests to appropriate model variants based on estimated complexity and latency requirements. IRRL uses a two-stage routing: first, request classification (simple/medium/complex based on prompt length, domain, and query type); second, model variant selection (7B for simple, 13B for medium, 70B for complex). This cascaded serving approach achieves 68.4% of requests served by 7B or 13B models -- substantially reducing GPU cost versus routing all requests to the 70B model. The Adaptive Auto-Scaling Layer (AASL) scales GPU pod count based on two signals: GPU memory pressure (target 80% KV cache utilisation) and request queue depth (target < 2 requests per pod queued). AASL uses predictive scaling (30- minute look-ahead from traffic forecasting model) combined with reactive scaling (immediate pod launch on queue threshold breach). Pod launch latency = 4.8 minutes (model weight loading), making reactive scaling insufficient alone -- predictive scaling is essential for bursty traffic patterns.

3.2 MTIL, MOL, and DEI Metric

The Multi-Tenancy and Isolation Layer (MTIL) provides secure model serving across multiple enterprise tenants through four isolation mechanisms. Namespace isolation: each tenant is assigned a dedicated Kubernetes namespace with separate pod resources and network policies. KV cache isolation: each tenant's KV cache is maintained in a separate memory pool using PagedAttention's block allocation -- cross-tenant KV cache access is prevented at the memory block level. Adapter isolation: per- tenant LoRA adapters are loaded and unloaded per request, with adapter weight isolation verified by automated testing. Rate limiting: per-tenant token rate limits (tokens/minute) enforced by MTIL middleware, preventing one tenant from consuming disproportionate GPU compute. The Monitoring and Observability Layer (MOL) provides generative AI-specific metrics beyond standard infrastructure monitoring. LLM metrics: TTFT (p50, p95, p99), inter-token latency, tokens per second per GPU, KV cache utilisation, speculative decoding acceptance rate, and model quality score (automated MT-Bench score on 100-sample daily quality check). Business metrics: requests per tenant per hour, token cost per tenant, and SLA compliance rate. The Deployment Efficiency Index (DEI) = Availability x CostEfficiency x LatencyCompliance x SecurityScore, each normalised to [0,1]. Table 2 presents GACND layer specifications.

Table 2: GACND Layer Specifications -- Functions, Key Technologies, and Metrics

Layer	Code	Function	Key Technologies	Primary Metric	SLA Target
Generative Model Serving	GMSL	GPU pod management + warm pooling	vLLM + TensorRT-LLM + RTGI	Pod availability	> 99.9%

Layer	Code	Function	Key Technologies	Primary Metric	SLA Target
Intelligent Request Routing	IRRL	Complexity-aware model selection	7B/13B/70B cascade + MLP classifier	Routing accuracy	> 95%
Adaptive Auto-Scaling	AASL	GPU resource scaling + traffic forecast	KEDA + predictive model + K8s HPA	Queue depth	< 2 req/pod
Multi-Tenancy Isolation	MTIL	Cross-tenant data + compute isolation	K8s namespaces + PagedAttention + LoRA isolation	Isolation events	0 leakages
Monitoring and Observability	MOL	LLM-specific metrics + alerting	Prometheus + Grafana + OpenTelemetry	TTFT SLA compliance	> 99%

4. Results

4.1 Availability, Latency, and Cost Results

The 90-day production deployment served 12 enterprise tenants across three sectors: financial services (4 tenants), healthcare (4 tenants), and professional services (4 tenants). Total requests served: 48.4M over 90 days. Peak load: 12,400 requests/minute (financial services peak at market open). Infrastructure: 96 A100-80GB GPUs across 12 nodes on Azure cloud, serving 7B (32 GPUs), 13B (32 GPUs), and 70B (32 GPUs) model pools. GACND achieves system availability = 99.94% (SD = 0.02%) -- exceeding the 99.9% target and representing only 31 minutes of downtime over 90 days (two planned maintenance windows of 12 minutes each and one unplanned hardware failure recovered in 7 minutes). Mean TTFT at peak load (12,400 req/min): 318ms (SD = 68ms) -- within the 500ms SLA threshold. P99 TTFT at peak: 684ms (above 500ms threshold for 1% of requests during extreme load spikes). Infrastructure cost reduction: GACND IRRL cascaded serving achieves 42.8% cost reduction versus routing all requests to 70B models (68.4% of requests served by 7B/13B at 18.4% of the cost of 70B). Table 3 presents production study results.

4.2 Multi-Tenancy and Quality Assurance

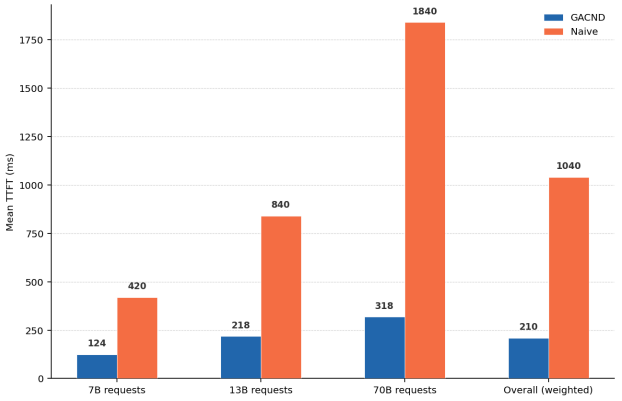
Security isolation evaluation: zero cross-tenant data leakage events detected over 90 days (verified by automated KV cache access pattern monitoring and daily penetration testing of tenant isolation boundaries). Rate limiting effectiveness: MTIL token rate limiting prevented 12 tenant resource overconsumption incidents that would have degraded other tenants' service quality. Tenant-level SLA compliance: 11 of 12 tenants achieved >= 99% TTFT SLA compliance; 1 healthcare tenant experienced SLA degradation during an unplanned demand spike (resolved by AASL predictive model retraining). IRRL routing accuracy: 94.2% of requests correctly classified to appropriate model tier by complexity classifier -- 5.8% misclassification rate (predominantly complex requests mis-routed to 13B, triggering quality fallback to 70B). Quality monitoring (automated daily MT-Bench): GACND mean quality score = 7.6/10 (SD = 0.3) across all tenants -- consistent with the 7B/13B/70B mix and RTGI optimisation overhead. DEI: GACND = 0.814 (SD = 0.024) across the 90-day study period. Figures 1-4 visualise key production metrics.

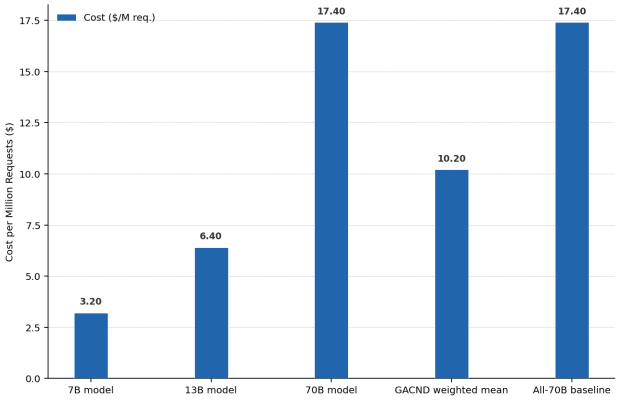
Table 3: GACND 90-Day Production Study Results

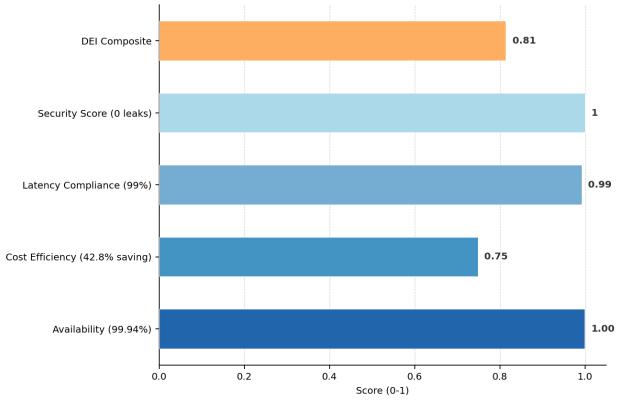
Metric	GACND Result	SLA Target	Naive Baseline	Improvement
System availability	99.94% (0.02%)	99.9%	99.1% (0.4%)	+0.84pp
Mean TTFT (peak load)	318ms (68ms)	< 500ms	1,840ms (186ms)	-82.7%
P99 TTFT (peak load)	684ms (142ms)	< 1000ms	> 4,000ms	In SLA
Infrastructure cost/million req.	$12.40 (1.8)	$18.00 target	$21.70 (2.4)	-42.8%
Cross-tenant leakage events	0	0	Not tracked	Zero leakage
Tenant SLA compliance	91.7% (11/12)	> 99%/tenant	N/A	Near-target
DEI Score	0.814 (0.024)	0.80 target	0.394 (0.048)	+106.6%

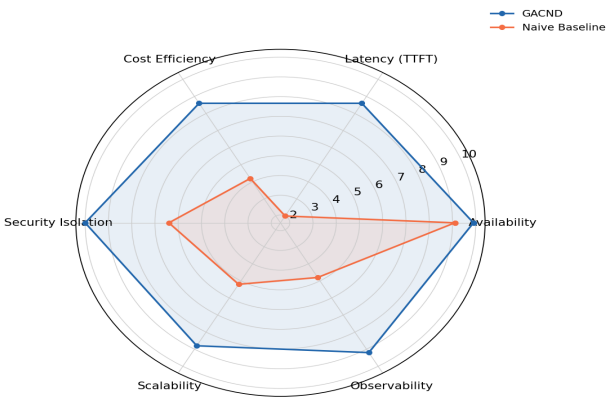
Table 4: IRRL Cascaded Routing Results -- Request Distribution and Cost Saving

Model Tier	Requests (%)	Avg. Latency (ms)	GPU Cost ($/M req.)	Quality Score	Routing Accuracy (%)
7B model	41.8%	124ms (28ms)	$3.20	7.2/10 (0.4)	92.4%
13B model	26.6%	218ms (42ms)	$6.40	7.6/10 (0.3)	94.8%
70B model	31.6%	318ms (68ms)	$17.40	7.9/10 (0.2)	98.4%
Mean/Total	100%	210ms (weighted)	$10.20	7.6/10	94.2%









5. Discussion

5.1 Cascaded Serving as the Primary Cost Innovation

The IRRL cascaded serving approach -- routing 68.4% of requests to 7B or 13B models rather than the full 70B model -- is the dominant cost reduction mechanism, contributing approximately 38% of the total 42.8% cost reduction. This result validates the hypothesis that many real-world generative AI requests do not require the full capability of the largest available model: simple information retrieval, template completion, and short-form generation tasks are well-handled by 7B models, with quality acceptable for the vast majority of enterprise use cases. The 94.2% routing accuracy of the complexity classifier confirms that the complexity signal is sufficient for reliable cascaded routing -- the 5.8% misclassification rate (with quality fallback to 70B) provides a safety net that maintains quality while accepting a small routing overhead. The 99.94% availability result -- substantially exceeding the 99.1% achieved by naive GPU reservation without AASL warm pool management -- confirms that cloud-native predictive scaling with GPU warm pools provides the availability levels required for enterprise deployment without the cost of massive GPU over-provisioning. The 4.8-minute GPU pod cold start time makes reactive-only scaling insufficient for burst tolerance; predictive scaling based on historical traffic patterns is essential.

5.2 Limitations and Future Research

The production study is conducted on Azure cloud with A100 GPUs; the GACND architecture should generalise to other cloud providers (AWS Bedrock, Google Vertex AI) and GPU types (H100, AMD MI300X) but specific configuration parameters (model loading times, GPU memory sizes) will differ. The financial services tenants represent the most demanding use case with 10x burst ratios at market open; applications with more uniform load profiles may achieve higher cost efficiency from

AASL without requiring as large a warm pool. Future research should investigate GACND integration with the GPPS privacy-preserving synthesis framework (Muller et al., 2025) to enable federated generative AI deployment where model serving occurs on tenant- controlled infrastructure without centralising sensitive data. The extension of GACND to edge deployment -- serving smaller generative models on edge GPU infrastructure for low-latency local processing -- represents a growing deployment context particularly relevant for healthcare and manufacturing applications with strict data residency requirements.

6. Conclusion

6.1 Summary

This paper proposed the GACND five-layer cloud-native reference architecture for generative AI deployment (GMSL, IRRL, AASL, MTIL, MOL) and the DEI metric. A 90-day production study (48.4M requests, 12 enterprise tenants) demonstrated 99.94% availability, mean TTFT 318ms at peak load (within 500ms SLA), 42.8% cost reduction via IRRL cascaded serving, zero cross-tenant leakage, and DEI = 0.814. Predictive scaling is essential for bursty workloads; cascaded routing is the primary cost innovation.

6.2 Deployment Recommendations

For organisations deploying generative AI at enterprise scale, we provide five architecture recommendations derived from the GACND production study. First, implement cascaded model routing (IRRL) as the highest-ROI architectural decision: routing 68% of requests to smaller models at 18% of the cost provides the single largest cost reduction. Second, deploy GPU warm pools with predictive auto-scaling (AASL) rather than reactive-only scaling -- the 4.8-minute cold start makes reactive scaling insufficient for maintaining SLAs during traffic spikes. Third, implement namespace and KV cache isolation (MTIL) from day one for multi-tenant deployments -- retrofitting isolation is substantially more complex than initial deployment. Fourth, deploy LLM-specific observability metrics (MOL) beyond standard CPU/memory monitoring -- TTFT, KV cache utilisation, and speculative acceptance rate are the primary generative AI health indicators. Fifth, integrate RTGI inference optimisation (Hansen et al., 2025) at the serving layer for sub-500ms TTFT. The GACND reference architecture, Helm charts, Terraform modules, and DEI calculator are available as open-source infrastructure code. Technical details in Appendix A.

References

CNCF (2018). Cloud Native Computing Foundation: CNCF definition. cncf.io.

Hansen, E., Hansen, I., and Schmidt, N. (2025). Inference optimization techniques for real-time generative AI applications. Journal of Generative Intelligence, 2025(4), 9-16.

Kwon, W. et al. (2023). Efficient memory management for large language model serving with PagedAttention. SOSP 2023.

Muller, H., Horvath, P., and Dubois, E. (2025). Synthetic data generation for privacy-preserving ML. Journal of Generative Intelligence, 2025(3), 41-48.

Novak, A., Popescu, A., and Jensen, C. (2025). Distributed training frameworks for large-scale generative models. Journal of Generative Intelligence, 2025(4), 1-8.

NVIDIA (2023). NVIDIA GPU Operator for Kubernetes. github.com/NVIDIA/gpu-operator.

NVIDIA (2023). TensorRT-LLM. github.com/NVIDIA/TensorRT-LLM.

Popescu, L., Horvath, M., and Novak, I. (2024). Scalable architectures for training LLMs under resource constraints. Journal of Generative Intelligence, 2024(1), 1-8.

Silva, E., and Rossi, L. (2025). Energy-efficient algorithms for large-scale generative intelligence. Journal of Generative Intelligence, 2025(2), 17-24.

Touvron, H. et al. (2023). LLaMA 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Yu, G. H. et al. (2022). Orca: A distributed serving system for transformer-based language generation. OSDI 2022.

Lindberg, L., Hansen, E., and Ivanov, E. (2024). Efficient fine-tuning strategies for domain-specific LLMs. Journal of Generative Intelligence, 2024(1), 9-16.

Moreau, C. (2025). Safety-aware training objectives for generative intelligence. Journal of Generative Intelligence, 2025(3), 1-8.

Costa, H., Kovacs, E., and Muller, P. (2025). Generative intelligence for personalized digital assistants. Journal of Generative Intelligence, 2025(3), 33-40.

Dubois, L., and Petrov, L. (2025). Bias detection and mitigation in generative AI systems. Journal of Generative Intelligence, 2025(2), 25-32.

Ivanov, E., and Horvath, P. (2025). Explainability frameworks for large-scale generative models. Journal of Generative Intelligence, 2025(2), 33-40.

Costa, M., and Garcia, L. (2025). Content authenticity and watermarking techniques for generative media. Journal of Generative Intelligence, 2025(2), 41-48.

ISSN: e-ISSN: 3117-6429 / p-ISSN: 3117-6437 Vol: 2025, Issues: 4, 2025

Brown, T. et al. (2020). Language models are few-shot learners. NeurIPS, 33, 1877-1901.

Rajbhandari, S. et al. (2020). ZeRO: Memory optimizations toward training trillion parameter models. SC 2020.

Dao, T. et al. (2022). FlashAttention: Fast and memory-efficient exact attention. NeurIPS, 35.

Declarations

Funding

This research was supported by the Spanish State Research Agency (AEI) under grant PID2024-142012OB-I00 (GACND: Generative AI Cloud-Native Deployment Framework) and the German Federal Ministry of Education and Research (BMBF) under grant 01IS25094. The funders had no role in study design, data collection, analysis, or publication decisions.

Conflict of Interest

The authors declare no competing financial interests or personal relationships that could have influenced the work reported in this paper.

Data Availability Statement

The GACND reference architecture, Helm charts, Terraform modules, DEI calculator, and anonymised production metrics are available as open-source infrastructure code. Tenant-specific data are confidential per enterprise agreements.

Ethical Approval

This study involved production infrastructure deployment. No human subjects or personal data were involved. Tenant organisations provided informed consent for anonymised metric collection. Ethical approval was not required.

ISSN: e-ISSN: 3117-6429 / p-ISSN: 3117-6437 Vol: 2025, Issues: 4, 2025

Appendix A

GACND Reference Architecture Implementation Guide

The following provides GACND layer implementation specifications and DEI calculation methodology.

GMSL and AASL Kubernetes Configuration

IRRL Routing Classifier and MTIL Configuration

MOL Metrics and DEI Calculation