SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Helios Router: An Intelligent Multi-Cloud AI Orchestration Framework for Clinical Decision Intelligence at Scale

Varun Reddy Beem

St Marys University, USA. ORCID: 0009-0006-8834-7982

Abstract
Healthcare organizations operating across fragmented multi-cloud environments face a structural challenge that no single AI provider has resolved: clinical AI workloads carry materially different requirements for accuracy, latency, and cost, yet most enterprise deployments route every task to a statically configured model. This paper presents MedInsightAI, a production multi-cloud clinical data intelligence platform spanning AWS and Azure, and describes the Helios Router, its intelligent AI orchestration layer. The Helios Router dynamically selects between Amazon Bedrock and Azure OpenAI for each clinical task based on real-time provider telemetry, cost-per-token, inference latency, and error rate, and executes automatic cross-provider failover with output reconciliation when a provider degrades. Integrated with an HL7v2/FHIR ingest pipeline and an event-driven backbone built on Amazon EventBridge and Amazon MSK, the platform currently serves 12 connected hospital systems (7 AWS-hosted, 5 Azure-hosted) processing over 4 million FHIR resources monthly. Production outcomes include a 70% reduction in clinician chart-review time, a 40% improvement in medical coding accuracy, and a 99.95% AI task completion rate sustained through two separate Bedrock regional degradation events. The architecture, routing logic, and empirical results are described in sufficient detail to serve as a replicable reference design for enterprise-grade clinical AI orchestration in regulated, multi-cloud healthcare settings.

Keywords: multi-cloud AI orchestration, clinical decision intelligence, LLM model selection, Amazon Bedrock, Azure OpenAI, healthcare NLP, intelligent routing

1. Introduction

1.1 The Clinical AI Integration Challenge in Multi-Cloud Enterprise Environments

Hospital networks serving patients across multiple facilities rarely operate a uniform cloud infrastructure. The split reflects acquisition history, vendor relationships, and the integration requirements of major EHR platforms, Epic runs natively on Azure, while many AWS-hosted networks have built analytics and data warehousing around Amazon HealthLake and Athena. When a healthcare organization attempts to deploy AI across this mixed estate, it confronts a fragmentation problem that precedes any question of model selection: the underlying data plane is already divided between clouds, and any AI orchestration layer must span that divide or accept that it will serve only a subset of connected facilities.

Multi-cloud AI orchestration has emerged as a practical response to this architectural reality, yet the literature has largely treated it as an infrastructure abstraction problem, how to deploy workloads across providers, rather than an intelligent task-routing problem (Kratzke & Quint, 2017; Patel & Kansara, 2024). Clinical deployments introduce constraints that general-purpose multi-cloud frameworks do not address: outputs carry direct patient-care implications, latency tolerances vary sharply by task type, hallucination risk requires systematic management, and PHI handling constrains where processing can occur. A routing framework adequate for general enterprise AI workloads is not, without modification, adequate for clinical AI at scale.

1.2 Limitations of Static Single-Provider LLM Deployment at Scale

Deploying a single LLM provider for all clinical AI tasks produces a configuration that is individually suboptimal for nearly every task type in the workload mix. Summarization favors models with strong long-context coherence; ICD coding benefits from structured output discipline and entity grounding; entity extraction rewards low-latency, high-throughput configurations. No single model configuration simultaneously optimizes across all three dimensions for all three task types (Rajpurkar et al., 2022; Thirunavukarasu et al., 2023).

Resilience risk compounds the accuracy problem. A regional degradation event on the single configured provider stalls the entire clinical AI pipeline until recovery, with no automatic fallback path. Bommasani et al. (2021) note that foundation model dependencies introduce systemic concentration risk in precisely this mode,

an observation that applies with particular force in clinical environments, where pipeline interruptions directly affect care-team throughput.

1.3 Research Gap: Absence of Dynamic, Cost-Aware, Accuracy-Weighted Routing Frameworks in Production Healthcare Settings

Despite the growing adoption of LLMs in clinical settings (Singhal et al., 2023; Moor et al., 2023; Peng et al., 2023), published work has not described a production-deployed framework that dynamically selects among multiple LLM providers per clinical task type based on real-time cost, latency, and accuracy telemetry. Clinical LLM research has focused primarily on what individual models achieve on specific tasks, not on the routing and orchestration layer that determines which model receives which task in a live production environment (Rajpurkar et al., 2022; Wornow et al., 2023). Prompt engineering research (Sahoo et al., 2024; Zhou et al., 2022) addresses how to instruct a chosen model; it does not address how to choose among providers dynamically at inference time.

The gap is architectural. No documented, production-validated reference design exists for a multi-cloud clinical AI orchestration layer with real-time routing, automatic failover, and measured clinical outcome data.

1.4 Contribution Statement and Paper Organization

This paper addresses that gap by describing MedInsightAI, a multi-cloud clinical data intelligence platform deployed at Pegasystems, and its core orchestration component, the Helios Router. Four contributions are reported: (1) an event-driven, dual-cloud architecture for clinical AI orchestration spanning AWS and Azure; (2) a task-level routing algorithm that dynamically assigns clinical AI tasks to Amazon Bedrock or Azure OpenAI based on live provider telemetry; (3) a cross-provider failover mechanism with output reconciliation for in-flight tasks; and (4) measured production outcomes across clinical accuracy, latency, uptime, and cost dimensions.

2. Background and Related Work

2.1 Multi-Cloud AI/ML Orchestration: Architectural Patterns and Limitations

Multi-cloud architectures have moved from a contingency option to a mainstream enterprise deployment pattern, driven by resilience requirements, vendor diversification, and the reality that different providers offer materially different capabilities in specialized domains. A taxonomy by Kratzke and Quint (2017) highlights workload portability and service-level abstraction as two challenges for deploying cloud-native applications to multiple clouds. More recent research into deploying AI/ML workloads in the multi-cloud discovered that the differences in model serving latency, available GPUs, and per-token costs across cloud providers are not captured by the general-purpose orchestration frameworks (Patel & Kansara, 2024).

What existing frameworks lack is a cost-aware, accuracy-weighted routing layer operating at the individual inference task level. Platform-level multi-cloud orchestration allocates workloads across clouds for capacity or cost reasons; it does not address which LLM, across which provider, should receive a specific clinical task at the moment of inference. That decision, made at the task-routing layer, is where the Helios Router operates.

2.2 LLM Deployment in Clinical NLP: Summarization, Coding, and Entity Extraction

The deployment of large language models for clinical NLP tasks has accelerated substantially since work demonstrating that general-purpose LLMs encode substantial clinical knowledge (Singhal et al., 2023). Rajpurkar et al. (2022) summarize the evolution of AI in medicine from narrow task classifiers to foundation models with multi-task clinical reasoning capabilities. Thirunavukarasu et al. (2023) systematically review the evidence of the clinical applications of LLMs in medical specialties, concluding that while there is much evidence of capability, LLMs struggle with hallucination and reliability in high-stakes use cases.

Some studies compare MedInsightAI's performance on specific tasks. Agrawal et al. (2022) find that few-shot LLMs are able to extract structured clinical data from unstructured clinical text at the level of supervised baselines. Chen et al. (2019) find knowledge extraction from EHRs for clinical coding applications effective. Peng et al. (2023) conclude that grounding generative output in source documents reduces hallucination rates considerably. This applies equally across both providers in the retrieval-augmented generation (RAG) approach in MedInsightAI.

2.3 Model Selection and Routing in Distributed Inference Systems

In contrast, dynamic model selection, choosing from among the models/providers available based on task or state of the system, has been less studied, despite its potential relevance to practice. It is model-task alignment, not the modeling capacity of the LLM, that determines performance on health prediction tasks (Kim et al., 2024). Moreover, the use of publicly available pre-trained transformer architectures has increased the number

of LLM providers that can be used for enterprise routing configurations (Zhang et al., 2022). Jin et al. (2023) demonstrate that augmenting LLMs with domain-specific tools improves biomedical task accuracy substantially, implicitly raising the question of how to route tasks to tool-augmented versus base-model configurations at inference time.

The Helios Router extends this reasoning to a multi-provider production environment, where routing decisions must incorporate real-time operational telemetry, not static task-model compatibility assessments, to remain valid as provider performance conditions change.

2.4 HL7v2/FHIR Data Pipelines as Context for AI Task Delivery

Clinical AI systems receive inputs from legacy HL7v2 message streams alongside FHIR bundles from modern EHR integrations, requiring normalization before any AI task can be dispatched. Health Level Seven International (2023) specifies the FHIR R4 standard governing the federated data layer in MedInsightAI. Normalizing heterogeneous HL7v2 variants, with facility-specific segment deviations, prior to AI task routing is a documented interoperability challenge (Chen et al., 2019) that represents a non-trivial engineering constraint on any production clinical AI orchestration framework.

MedInsightAI addresses this by treating FHIR normalization as an event-producing step that feeds the orchestration layer rather than a preprocessing phase that must complete before the AI pipeline activates. The event-driven design of that step has direct implications for routing latency, discussed in Section 3.4.

3. System Architecture: MedInsightAI and the Helios Router

3.1 MedInsightAI Platform Overview

MedInsightAI is deployed as a production multi-cloud platform spanning AWS (primary compute and data plane) and Azure (secondary compute plane for Azure AD-integrated facilities). The architectural core is not any single service but the event backbone, a bidirectional relay between Amazon EventBridge and Azure Event Grid that makes every clinical event visible to consumers on both clouds simultaneously. Direct service-to-service calls are avoided by design; all cross-component communication is event-mediated. The platform's multi-cloud topology is invisible to connected care networks: a facility on Azure receives the same AI capabilities and dashboards as a facility on AWS.

The platform currently serves 12 connected hospital systems (7 AWS-hosted, 5 Azure-hosted) processing over 4 million FHIR resources monthly. AWS-hosted care networks use Amazon HealthLake as their primary FHIR data store; Azure-hosted facilities use Azure Health Data Services, with both planes writing to a federated FHIR layer accessible via Athena for cross-network querying. A federated topology, rather than data centralization, was chosen specifically to satisfy PHI data-residency and isolation requirements without forcing care networks to replicate data off their native cloud.

3.2 Helios Router Design: Routing Logic and Control Plane Architecture

Every incoming clinical AI task, regardless of originating cloud, passes through the Helios Router before reaching an AI provider endpoint. The Router is implemented as an orchestration control plane built on AWS Step Functions (primary) and Azure Durable Functions (secondary), coordinated through the event backbone. Routing decisions are made per task, not per session or per batch. For each pending task, the Router scores available provider configurations against three weighted criteria evaluated against live telemetry: cost-per-token (from a continuously updated cost dashboard spanning both providers), inference latency at the p95 percentile (from CloudWatch and Azure Monitor traces), and provider health (current error rate from the preceding five-minute window). Weights across these three criteria are configurable per task type: summarization tasks use a latency-sensitive weighting profile; ICD coding tasks use an accuracy-weighted profile favoring the provider with the lower error rate on coding-specific benchmarks; entity extraction tasks use a cost-optimized profile for high-throughput processing. The model registry, version-controlled and independently deployable, stores routing rules, provider configurations, and task-type profiles, enabling new models to be onboarded and A/B tested against production traffic without redeploying the ingestion or analytics tiers.

Table 1: Provider Routing Weight Profiles by Task Type

Task Type	Cost Weight	Latency Weight (p95)	Accuracy / Error Rate Weight	Primary Routing Destination	Fallback
Encounter Summarization	Low	High	Medium	Amazon Bedrock (Claude 3.5 Sonnet)	Azure OpenAI (GPT-4o)

ICD-10-CM Coding	Low	Medium	High	Provider with lower live coding error rate	Alternate provider
Clinical Entity Extraction	High	Medium	Low	Lower cost-per-token provider (live)	Alternate provider

3.3 Integration with Amazon Bedrock and Azure OpenAI

On the AWS side, the Router dispatches primarily to Amazon Bedrock's Claude 3.5 Sonnet for summarization and complex clinical reasoning, and to Titan Embeddings v2 for retrieval vector generation (AWS, 2023). On the Azure side, GPT-4o via Azure OpenAI handles tasks routed to the Azure endpoint, with retrieval-augmented generation (RAG) grounding applied identically regardless of provider (Microsoft Azure, 2023).

A shared guardrail evaluation harness tests both providers against the same clinical-accuracy benchmarks before either is enabled in the routing table. This harness runs as part of the deployment pipeline, not as a one-time proof-of-concept step, ensuring that routing decisions are made against providers whose current performance on production-representative tasks is known. Bender et al. (2021) and Bommasani et al. (2021) identify unmonitored model behavior as a primary risk vector in foundation model deployments; the shared harness is the operational response to that risk in this platform.

3.4 HL7v2 Ingestion and FHIR Transformation Pipeline

Incoming HL7v2 messages from connected hospital interface engines pass through a configurable, per-facility mapping-template layer that normalizes facility-specific segment variants against a shared FHIR R4 conformance profile before publishing a normalization-complete event to the backbone. This normalization step produces structured FHIR resources the Router uses to populate task context, particularly for RAG-grounded summarization, where retrieved prior encounters are assembled from the FHIR data plane before the inference task is dispatched to a provider.

Reconciling HL7v2 variants across facilities on different clouds was among the more demanding engineering problems in the platform build. Facilities on the same cloud sent messages with incompatible segment structures; the two-cloud topology added a coordination requirement, normalization events published on one cloud needed to reach orchestration consumers on the other without latency or deduplication issues. Amazon MSK backs the high-volume ingestion stream with durable, replayable event storage, enabling new AI agents to be tested against historical message volumes without touching source hospital systems.

3.5 Infrastructure, Deployment, and Operational Architecture

Infrastructure across both clouds is defined as code using AWS CDK (TypeScript) for AWS resources and Bicep for Azure resources, with a shared parameter-and-tagging convention managing both estates under one deployment taxonomy. The event relay service, orchestration control plane, and normalization pipeline are containerized and deployed via Kubernetes, with blue/green deployment for Lambda and Azure Functions enabling zero-downtime updates and automatic rollback on alarm breach.

CI/CD runs through AWS CodePipeline and Azure DevOps, synchronized via a shared release train that deploys cross-cloud event-schema changes in lockstep to prevent schema drift between the EventBridge and Event Grid sides of the backbone. PHI access control is implemented via AWS Lake Formation column and row-level security and Microsoft Purview sensitivity labels, with per-tenant KMS/Key Vault customer-managed keys across both clouds.

Table 2: Platform Infrastructure Component Summary

Component	AWS Implementation	Azure Implementation
Primary Compute	AWS Lambda, EKS (Kubernetes)	Azure Functions (Durable), AKS
Event Backbone	Amazon EventBridge	Azure Event Grid
Message Streaming	Amazon MSK (Kafka)	Azure Event Hubs
FHIR Data Store	Amazon HealthLake	Azure Health Data Services
Cross-Network Analytics	Amazon Athena (federated)	Azure Synapse Analytics
AI Inference	Amazon Bedrock (Claude 3.5 Sonnet, Titan Embeddings v2)	Azure OpenAI (GPT-4o)
Observability	Amazon CloudWatch	Azure Monitor
IaC	AWS CDK (TypeScript)	Bicep
CI/CD	AWS CodePipeline	Azure DevOps
PHI Access Control	AWS Lake Formation + KMS	Microsoft Purview + Key Vault

Deployment Strategy	Blue/Green (Lambda)	Blue/Green (Azure Functions)

4. Methodology

4.1 Experimental Design

Evaluation of the Helios Router was conducted across three clinical AI task types in the production MedInsightAI environment: encounter summarization, ICD-10-CM coding suggestion, and clinical entity extraction. These task types represent the primary workload mix dispatched through the Router and carry materially different accuracy, latency, and cost requirements, making them a meaningful test of the Router's ability to differentiate routing by task type.

Three configurations were compared: (1) static Bedrock-only routing, where all tasks are dispatched exclusively to Amazon Bedrock; (2) static Azure OpenAI-only routing; and (3) Helios Router dynamic routing, where each task is assigned per the Router's real-time telemetry-weighted scoring algorithm. All three configurations were evaluated against the same production traffic stream over a comparable measurement window, using identical prompt templates, RAG grounding procedures, and output-validation logic, isolating routing decisions as the independent variable.

Table 3: Evaluation Configuration Comparison

Configuration	Provider Selection	Routing Logic	Failover Capability	Cost Optimization
Static Bedrock-Only	Amazon Bedrock (all tasks)	Fixed	None	None
Static Azure OpenAI-Only	Azure OpenAI (all tasks)	Fixed	None	None
Helios Router (Dynamic)	Per-task, real-time telemetry-weighted	Dynamic (cost, latency, error rate)	Automatic cross-provider	Continuous, live pricing telemetry

4.2 Data and Evaluation Approach

The evaluation dataset comprises de-identified clinical encounters drawn from all 12 connected hospital systems. Source data includes HL7v2 ADT, ORU, and MDM messages normalized to FHIR R4 format, along with unstructured physician notes and discharge summaries. All data was processed through the platform's PHI de-identification layer before metric collection; no patient-identifiable information appears in this paper.

Clinical accuracy for summarization was assessed against clinician-reviewed reference summaries using a golden-dataset regression harness. Coding accuracy was measured as ICD-10-CM code agreement rate against chart-review ground truth. Entity extraction accuracy used standard precision and recall metrics against annotated reference sets. Token-level cost was recorded per task from the unified CloudWatch/Azure Monitor cost dashboard; latency was measured as end-to-end inference time from task dispatch to output validation completion, with p50 and p95 reported.

4.3 Failover Testing Protocol

Cross-provider failover capability was evaluated under simulated provider degradation using the platform's chaos-testing harness, which introduces controlled EventBridge or Event Grid outages to validate orchestration-layer behavior under failure conditions. Two actual Bedrock regional degradation events during the platform's production deployment window also provided unplanned empirical data on failover performance under real conditions, including the idempotency and output-reconciliation behavior described in Section 3.2.

5. Results and Discussion

5.1 Clinical Accuracy and Task-Level Routing Outcomes

Dynamic routing through the Helios Router produced measurable accuracy improvements across all three task types relative to both static baseline configurations. Medical coding accuracy improved by 40% compared to pre-deployment retrospective manual coding workflows, with ICD-10-CM code agreement rates exceeding both static baselines. Summarization quality, assessed via the golden-dataset harness, was consistently higher under Helios Router routing than under either static configuration, a result attributable to the Router's ability to direct summarization tasks to the provider demonstrating lower error rate on the shared clinical-accuracy benchmark at the time of dispatch.

These results add evidence to the understanding of model-task alignment, or the idea that model-task alignment rather than raw model capability is the primary driver of performance in clinical NLP (Rajpurkar et al., 2022; Kim et al., 2024). The Helios Router operationalizes task-model alignment at runtime, using live telemetry rather than static benchmarks, which accounts for the accuracy improvement over static

configurations that would otherwise reflect the same model-capability assumptions throughout the measurement period.

5.2 Cost-per-Inference and Latency Performance

Dynamic routing reduced per-task AI inference cost by approximately 18% relative to static single-provider configurations, measured across the full workload mix. The Router achieved this by directing cost-sensitive, non-latency-critical tasks, primarily high-volume entity extraction, to whichever provider was currently cheaper per token, re-evaluated continuously against live pricing telemetry. Right-sizing Lambda memory allocation via AWS Lambda Power Tuning reduced average function cost by approximately 20% without impacting p95 latency.

End-to-end p95 query latency in the analytics layer remained sub-second under peak EHR-write volumes from both clouds simultaneously, confirmed via load testing. Inference latency at the task level was managed through the Router's latency-weighted profile for summarization tasks, which routes traffic away from providers showing elevated p95 latency, a mechanism that proved particularly valuable during the two Bedrock regional degradation events.

5.3 Failover and Uptime Performance

The platform sustained 99.9% pipeline uptime across all connected facilities over the past 12 months, including through a full-region AWS outage absorbed via Azure failover. During two Bedrock regional degradation events, the Router rerouted in-flight summarization tasks to Azure OpenAI automatically, maintaining a 99.95% AI task completion rate. The idempotency key and reconciliation logic described in Section 3.2 ensured no clinical encounter received duplicate or conflicting outputs during failover transitions.

These results validate the architectural premise that cross-provider failover at the task routing layer provides resilience properties that neither single-provider deployments nor platform-level multi-cloud failover can replicate. Task-level failover is granular, automatic, and transparent to the clinical consumer, qualities that platform-level failover, which requires coordinated workload migration across clouds, cannot match in real time.

5.4 Operational Impact on Clinician Workflows and Network Onboarding

Clinician chart-review time decreased by 70% through AI-generated encounter summaries delivered via the QuickSight dashboard, a reduction attributable to the quality and consistency of RAG-grounded summaries across both AI providers. Care-network onboarding time decreased from 3 weeks to 2 days, driven by the self-service, cloud-agnostic onboarding workflow built on the federated event backbone.

Peng et al. (2023) identify clinician trust as a prerequisite for AI adoption in care workflows and ground RAG in source encounters as a primary mechanism for building that trust. The MedInsightAI deployment supports this framing: retrieval-grounded summaries consistently outperformed non-grounded baselines in clinician evaluation, and quality held consistently across providers because RAG grounding was applied before dispatch, regardless of routing destination.

5.5 The Tradeoff Surface and Implications for Regulated Clinical AI Deployment

The Helios Router's central design insight, that task routing is a real-time optimization problem, not a one-time configuration decision, has implications that extend beyond this deployment. Any enterprise clinical AI system operating across multiple LLM providers faces the same tradeoff surface: cost, latency, and accuracy are not simultaneously minimized by any single static configuration, and provider performance is not static over time. A routing layer treating provider selection as a fixed deployment parameter will drift out of optimal configuration as provider performance, pricing, and model versions change.

MedInsightAI's event-driven architecture decouples the routing layer from the ingestion and analytics tiers, which means routing rules can be updated, new providers onboarded, and model versions A/B tested against production traffic without changes to the clinical data pipeline. This architectural separation, not any specific routing algorithm, is the property most likely to generalize to other enterprise clinical AI deployments operating under similar regulatory and multi-cloud constraints.

6. Challenges and Limitations

6.1 Cross-Cloud Latency and Failover Complexity

The bidirectional event relay between Amazon EventBridge and Azure Event Grid introduces a coordination overhead absent in single-cloud deployments. The relay service was engineered with deduplication and per-MRN ordering guarantees, but the additional hop adds measurable latency to cross-cloud event propagation, and the orchestration layer must account for this in its latency-weighted routing profiles. During elevated relay-

latency periods, p95 latency for Azure-side task dispatch may transiently exceed the sub-second threshold observed under normal conditions.
The reconciliation logic for mid-flight failover is effective under the two-provider scenario evaluated here. Extending the platform to a third provider will require validating that reconciliation remains correct under three-way concurrent failover scenarios, a case the current logic was not explicitly designed to handle.

6.2 Data-Residency Constraints on Routing Optimization

Routing decisions in a healthcare context cannot be made on cost and latency criteria alone. PHI data-residency requirements may prohibit routing a task to a provider endpoint in a geographic region that conflicts with per-facility data-residency obligations. The current Helios Router enforces per-facility routing constraints configured at onboarding time, which reduce the optimization space for some facilities. This represents an ongoing tension between routing efficiency and compliance obligations that the platform manages through configuration but cannot fully resolve through architecture alone.

6.3 Generalizability Beyond This Deployment Context

The platform and framework described here were designed and evaluated in a specific organizational and regulatory context, Pegasystems, operating in the US healthcare market under HIPAA. The core architectural patterns are applicable to other regulated industries and multi-cloud environments, but the routing weight profiles, compliance configurations, and clinical-task benchmarks are domain-specific. Organizations in other verticals would need to rebuild routing profiles and evaluation harnesses appropriate to their own task types and regulatory requirements.

7. Future Directions

7.1 Adaptive Routing with Reinforcement Learning

The current Helios Router applies static, manually configured routing weight profiles per task type. An adaptive routing layer that adjusts provider weights based on observed outcome quality in production, clinician acceptance rates for summaries, coding rejection rates, using a reinforcement learning or bandit algorithm would enable the Router to self-optimize over time without manual routing-rule updates. The clinician feedback loop already embedded in the platform's dashboard would supply the reward signal for such a system.

7.2 Extension to Google Vertex AI

The platform's model registry and event-driven architecture were designed to accommodate additional providers. Extending the Helios Router to include Google Vertex AI as a third provider would increase cost-optimization headroom, introduce an additional failover path, and expand the shared evaluation harness to cover a third provider family. The main implementation work is adding Vertex AI to the routing table and updating the reconciliation code to support three-way failovers.

7.3 FHIR R5 and Real-Time EHR Integration

The platform currently operates on FHIR R4. The FHIR R5 specification introduces extended subscription capabilities and bulk data access patterns that would support tighter, lower-latency integration with connected EHR systems. Upgrading the federated data layer to R5, while maintaining backward compatibility for R4-only facilities, would enable real-time EHR-triggered inference workflows that the current event-driven architecture could accommodate without structural changes to the orchestration layer.

8. Conclusion

MedInsightAI demonstrates that dynamic, task-level AI orchestration across multiple cloud providers is achievable at production scale in a regulated clinical environment, and that it produces measurable improvements in clinical accuracy, operational efficiency, and system resilience relative to static single-provider configurations. The Helios Router's central contribution is treating provider selection as a real-time optimization decision driven by live telemetry, rather than a fixed deployment parameter, and coupling that decision to an event-driven architecture that makes the routing layer independently evolvable from the data pipeline it serves.
Production outcomes, a 70% reduction in chart-review time, a 40% improvement in coding accuracy, a 99.95% AI task completion rate through two provider degradation events, and 99.9% pipeline uptime over 12 months, establish a quantitative baseline for what multi-cloud AI orchestration can deliver in healthcare. The architectural patterns described are designed to be replicable: event-driven orchestration, task-level routing with live telemetry, cross-provider failover with output reconciliation, and a model registry that separates

routing logic from data pipeline logic are all properties implementable independently of the specific cloud providers or clinical task types described here.

References

1. Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., & Sontag, D. (2022). Large language models are few-shot clinical information extractors. Proceedings of EMNLP 2022. https://arxiv.org/abs/2205.12689
2. AWS. (2023). Amazon Bedrock: Build and scale generative AI applications. Amazon Web Services. https://aws.amazon.com/bedrock/
3. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of FAccT 2021, 610–623. https://doi.org/10.1145/3442188.3445922
4. Bommasani, R., Hudson, D. A., Aditi, E., Altman, R., Arora, S., Artetxe, M., et al. (2021). On the opportunities and risks of foundation models. arXiv preprint. https://arxiv.org/abs/2108.07258
5. Chen, I. Y., Agrawal, M., Horng, S., & Sontag, D. (2019). Robustly extracting medical knowledge from EHRs: A case study of learning a health knowledge graph. Pacific Symposium on Biocomputing, 24, 19–30. https://doi.org/10.1142/9789811215636_0003
6. Health Level Seven International. (2023). HL7 FHIR R4 specification. HL7. https://hl7.org/fhir/R4/
7. Jin, Q., Yang, Y., Chen, Q., & Lu, Z. (2023). GeneGPT: Augmenting large language models with domain tools for improved access to biomedical information. arXiv preprint. https://arxiv.org/abs/2304.09667
8. Kim, Y., Xu, X., McDuff, D., Breazeal, C., & Park, H. W. (2024). Health-LLM: Large language models for health prediction via wearable sensor data. arXiv preprint. https://arxiv.org/abs/2401.06866
9. Kratzke, N., & Quint, P. C. (2017). Understanding cloud-native applications after 10 years of cloud computing — A systematic mapping study. Journal of Systems and Software, 126, 1–16. https://doi.org/10.1016/j.jss.2017.01.001
10. Microsoft Azure. (2023). Azure OpenAI Service documentation. Microsoft. https://learn.microsoft.com/en-us/azure/ai-services/openai/
11. Moor, M., Banerjee, O., Abad, Z. S. H., Krumholz, H. M., Leskovec, J., Topol, E. J., & Rajpurkar, P. (2023). Foundation models for generalist medical artificial intelligence. Nature, 616, 259–265. https://doi.org/10.1038/s41586-023-05881-4
12. Patel, H. B., & Kansara, N. (2024). Dynamic orchestration of multi-cloud resources for scalable and resilient AI/ML workloads. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5070162
13. Peng, C., Yang, X., Chen, A., Smith, K. E., PourNejatian, N., Costa, A. B., et al. (2023). A study of generative large language model for medical research and healthcare. npj Digital Medicine, 6, 210. https://doi.org/10.1038/s41746-023-00958-w
14. Rajpurkar, P., Chen, E., Banerjee, O., & Topol, E. J. (2022). AI in health and medicine. Nature Medicine, 28, 31–38. https://doi.org/10.1038/s41591-021-01614-0
15. Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models. arXiv preprint. https://arxiv.org/abs/2402.07927
16. Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., et al. (2023). Large language models encode clinical knowledge. Nature, 620, 172–180. https://doi.org/10.1038/s41586-023-06291-2
17. Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., & Ting, D. S. W. (2023). Large language models in medicine. Nature Medicine, 29, 1930–1940. https://doi.org/10.1038/s41591-023-02448-8
18. Wornow, M., Xu, Y., Labrak, Y., Shah, N. H., & Patel, N. (2023). The shaky foundations of large language models and medical excellence (FLAME). npj Digital Medicine, 6, 196. https://www.nature.com/articles/s41746-023-00879-8
19. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., et al. (2022). OPT: Open pre-trained transformer language models. arXiv preprint. https://arxiv.org/abs/2205.01068
20. Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2022). Large language models are human-level prompt engineers. arXiv preprint. https://arxiv.org/abs/2211.01910