

Mapping the Open Agentic AI Landscape

From Open Models to Cross-Foundation Architectures



Vincent Caldeira

CTO APAC, Red Hat

vincent.caldeira@redhat.com



PHASE 01



Predictive ML

- Isolated and static analytical models
- Structured, relational tabular data pipelines
- Stateless, synchronous inference calls
- Simple containerized serving endpoints

PHASE 02



Generative Inference

- Massive 70B+ parameter transformer models
- High-throughput multi-modal input processing
- Distributed hardware scheduling across GPU clusters
- Streaming token responses requiring low TTFT

PHASE 03



Autonomous Agents

- Non-deterministic, dynamic execution flows
- Long-running, stateful session execution
- Multi-hop reasoning with persistent memory
- Dynamic real-world external tool usage via APIs



Core Architectural Insight: The AI landscape is now an end-to-end operational paradigm spanning data processing, fine-tuning, HPC, and multi-agent systems. Traditional stateless microservice architectures are fundamentally insufficient.

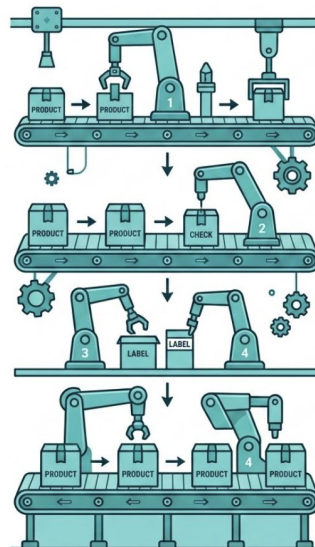


Traditional Apps

DETERMINISTIC

Request → Process → Database → Response

- ✓ **Deterministic Logic**
Linear, rule-based execution paths
- ✓ **Stateless Microservices**
Ephemeral compute, isolated states
- ✓ **Header-based Routing**
Deterministic HTTP path/header dispatch
- ✓ **Pre-defined Execution**
Well-bounded runtime requirements

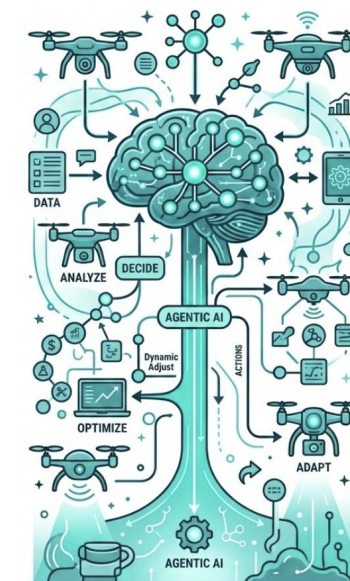


Agentic Workloads

AUTONOMOUS & STATEFUL

Memory ↔ Perception ↔ Reasoning ↔ Tool Action

- ★ **Probabilistic Behavior**
Dynamic multi-step reasoning chains
- ★ **Stateful & Long-Running**
Persistent context, memory retention
- ★ **Payload-based Routing**
Dynamic intent & semantic token routing
- ★ **Autonomous Tool Execution**
External API calls & runtime code generation





Layer 3: Agentic Interaction & Governance (AAIF Standards) [MCP](#) [A2A Protocol](#) [AGENTS.md](#) [Goose](#)

- Standardizes agent-to-tool (MCP) and agent-to-agent (A2A) interactions under neutral Linux Foundation governance [AAIF].
- Decouples agent cognitive reasoning from proprietary host environments and vendor APIs.



Layer 2: Cloud-Native Substrate & Governance (CNCf Orchestration) [Kubernetes](#) [KServe](#) [Istio Fabric](#) [Envoy AI Gateway](#) [vCluster / DRA](#)

- Offloads operational tasks—token rate-limiting, KV-cache routing, and confidential sandboxing—from Python app code into platform primitives.
- Enforces zero-trust workload identity (SPIFFE) and multi-tenant GPU isolation.

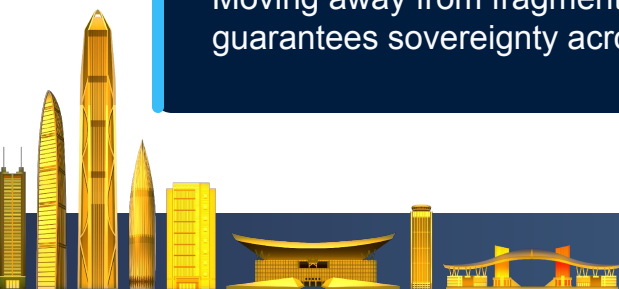


Layer 1: Distributed Compute & Hardware Abstraction (PyTorch Foundation) [PyTorch Core](#) [Ray](#) [vLLM Engine](#) [Triton](#) [OpenReg](#)

- Unifies model training, distributed orchestration, and high-throughput inference serving under open governance.
- Delivers cross-architecture enablement (NVIDIA CUDA, AMD ROCm, Intel, NPUs) to eliminate silicon lock-in.

The Strategic Architecture Shift

Moving away from fragmented, ad-hoc Python wrappers and proprietary API rentals toward a connected, vendor-neutral operational fabric that guarantees sovereignty across any cloud and accelerator.



Multi-Project Open Governance

PyTorch Core vLLM Ray DeepSpeed Safetensors

- Unifies deep learning training, distributed orchestration, and high-throughput inference serving under neutral Linux Foundation governance.
- Serves as the de facto industry execution standard, ensuring day-0 compatibility for open-weight foundation models.

Hardware Abstraction & Kernel Compilers

OpenReg TritonHelion DSL vLLM-cpu

- **OpenReg Interface:** Out-of-tree device registration allowing new silicon accelerators to plug directly into PyTorch without custom code forks.
- **Portable Compilers:** Triton and Helion DSL deliver state-of-the-art kernel performance across heterogeneous hardware without rewriting model code.
- **CPU Execution:** vLLM-cpu enables inference on standard CPUs (Intel Xeon AMX) to operate within power-constrained data centers.

Heterogeneous Silicon Choice

NVIDIA CUDA AMD ROCm Intel XPU Google TPU Regional NPU

- Makes hardware selection a runtime deployment decision rather than an architectural constraint.
- Delivers day-0 enablement for diverse AI accelerators and domestic NPUs to preserve hybrid-cloud compute agility.

PyTorch Foundation Framework & Serving Stack

PyTorch • vLLM • Ray • DeepSpeed

Hardware Abstraction & Compiler Layer

OpenReg Interface • Triton • Helion DSL

Heterogeneous Accelerator Pool

NVIDIA CUDA • AMD ROCm • Intel XPU • Google TPU • Domestic NPUs



Layer 2 (Platform): Cloud-Native AI Substrate

GOSIM Shenzhen 2026

TIER 1

Agentic & Developer Portal

Declarative templating and automated CI/CD integration workflows for AI engineers.

Backstage

kagent

TIER 2

Traffic & API Governance

Model virtualization, token-aware rate limiting, and centralized JWT/IRSA credential exchange.

Envoy AI Gateway

Gateway API

TIER 3

MLOps & Delivery

End-to-end model training pipelines, cryptographic OCI signing, and immutable packaging.

Kubeflow

KServe

Harbor

Kitops

TIER 4

Dynamic Orchestration

Gang scheduling, topology-aware accelerator placement, distributed inference co-scheduling, and multi-tenant hierarchical queuing

Volcano

Kueue

llm-d

TIER 5

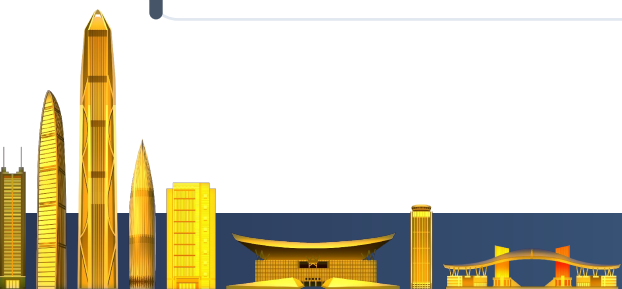
Execution & Hardware

Device-level GPU partitioning (MIG), kernel isolation, and hardware-enforced sandboxes.

NVIDIA GPU Operator

DRA

gVisor / Kata



Cloud Native Agentic Workload Definition — Microservice-oriented, containerized workloads on Kubernetes that leverage AI reasoning to plan, execute actions, and govern their own execution based on event-driven signals.

Pillar I

Identity & Tenancy

Workload Identity:

Cryptographically verifiable unique identities per agent.

Continuous Auth: Verify identity dynamically before every autonomous action.

Data Isolation: JIT access & query-level segregation on shared vector stores.

Pillar II

Security & Tools

Anti-Hijacking: Restrict execution to pre-approved schemas & boundaries.

Prompt Defense: Prevent system prompt leakage & sanitize internal flows.

Hardened Containers: Non-root execution & signed OCI provenance.

Pillar III

Observability

MELT Standard: Correlate telemetry metrics & traces for explainability.

AI Telemetry: Track GPU usage, token economics, & LLM Ops lifecycle.

Agent-as-Judge: Automated evaluation and tamper-resistant audit pipelines.

Pillar IV

Fault Tolerance

API Extensions: Path-based routing and custom inference endpoints.

K8s Resilience: Topology constraints & PodDisruptionBudgets.

Elastic Autoscaling: Dynamic HPA driven by inference & GPU pressure.

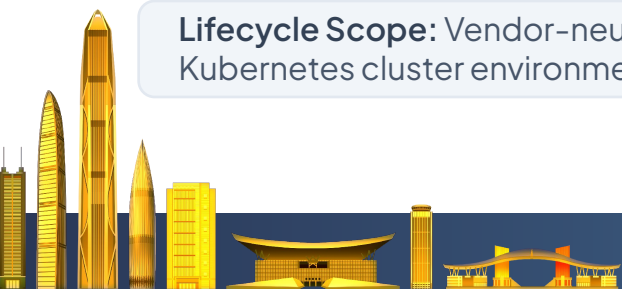
Lifecycle Scope: Vendor-neutral architectural governance across Kubernetes cluster environments

<> Develop

🔗 Distribute

📦 Deploy

▶ Runtime





GOVERNANCE & ECOSYSTEM

The AAIL Neutral Governance Layer

Neutral Ecosystem Home

AAIL (under Linux Foundation) stewards foundational open protocols:

MCP **A2A** AGENTS.md Goose

Vendor-Neutral Interoperability

Decouples agent reasoning from host infrastructure to prevent lock-in.

Prevents vendor lock-in across execution runtimes.



CORE OPEN PROTOCOLS

Protocol Architecture: MCP + A2A

A2A (AGENT-TO-AGENT)

HORIZONTAL LAYER



Discovers capabilities, negotiates tasks, & coordinates cross-domain workflows.

MCP (AGENT-TO-TOOL)

VERTICAL LAYER ('USB-C')



Universal standard connecting agents vertically to enterprise tools & data sources.

Protocol Synergy: A2A routes agent handoffs; MCP executes precise tool calls within agents.



VALUE & GOVERNANCE

Strategic Interoperability Outcome



Portable & Modular

Workloads interoperate across any framework, cloud, or enterprise boundary.

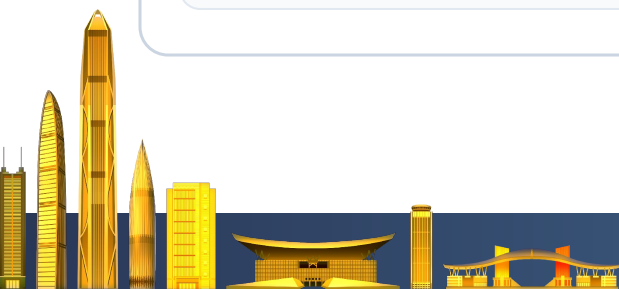
Cross-Framework Multi-Cloud



Standardized Governance

Uniform schema validation, identity mapping, and auditable tracing across all agent handoffs.

- Uniform schema validation
- Identity mapping & auth propagation
- End-to-end auditable tracing

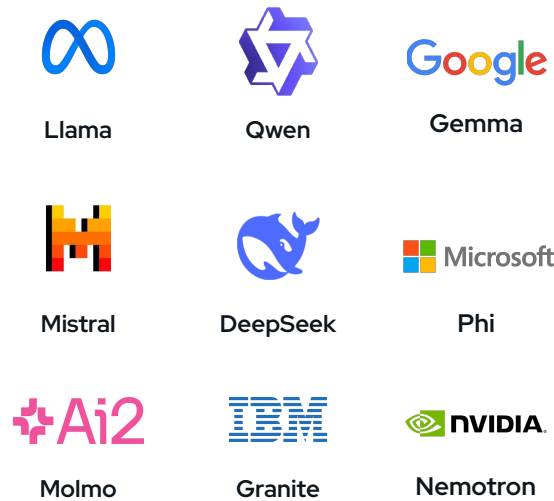
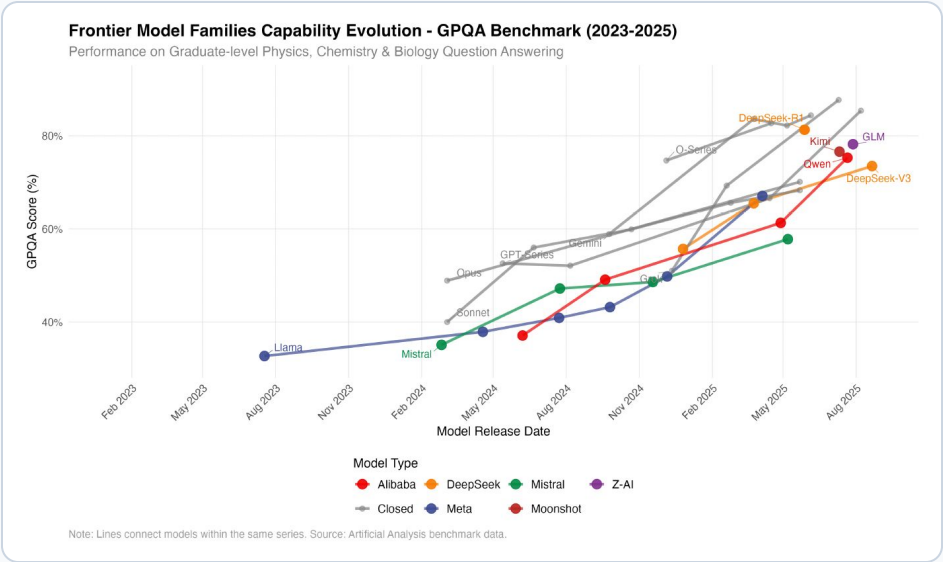


The Risk of Infrastructure Without Model Autonomy

- **Proprietary Monopoly:** Closed model APIs capture 96% of revenue and 80% of usage, exposing workloads to vendor lock-in.
- **Black-Box Opacity:** Users cannot inspect proprietary pipelines, creating risks around copyright, data provenance, and backdoors.
- **Cultural & Contextual Gap:** Mass-market models underperform on localized languages, regional regulations, and domain contexts.

The Convergence Advantage: Open Models Catch Up in 3–6 Months

Open-weight models match proprietary benchmark performance with a rapidly closing lag of only 3 to 6 months at up to 70%+ lower inference costs.



Open Model Validation

Continuously benchmark, curate, and optimize open models to deliver peak performance without API lock-in.



Supply Chain Integrity

Enforce verifiable provenance using OCI (ModelPack), SPDX 3.0 AI BOMs, and Sigstore signatures for tamper-proof integrity.



Domain Customization

Leverage post-training, fine-tuning, and RL loops via Training Hubs to adapt foundation models to local regulations and custom data.

Sources:
[The Latent Role of Open Models in the AI Economy](#) (Frank Nagle and Daniel Yue, 18 November 2025)
[Sovereign Large Language Models: Advantages, Strategy and Regulations](#) (Bondarenko et al., 15 January 2025)



Model Openness Framework (MOF)

Designed by LF AI & Data, this tiered rubric evaluates model completeness, requiring open training datasets, pipelines, and evaluation metrics.



Licensing with OpenMDW 1.1

Eliminating intellectual property and legal ambiguity by applying OSI-approved code licenses alongside purpose-built weights and parameter terms.



Standardized & Portable Packaging

We package models as standardized OCI artifacts using the ModelPack standard, making them portable and manageable like any cloud-native application.



AI Bill of Materials (AI BOM)

Enforcing automated, machine-readable tracking of training lineages, exact datasets, known packages, and supply chain dependencies via SPDX 3.0 for verifiable provenance.



Cryptographic Integrity

Utilizing Sigstore and Notary v2 to generate immutable, tamper-evident transparency logs (Rekor) to secure model weights from code to production to guarantee model origin and integrity.





Anchor on Open Models & Provenance

- **Eliminate black-box opacity** using open-weight models.
- **Enforce supply chain trust** via SPDX 3.0 AI BOMs & Sigstore signatures.



Offload Workload Control to Platform

- **Shift rate-limiting, KV-cache routing, and sandboxing** into CNCF primitives.
- **Enforce zero-trust SPIFFE workload identity** and confidential execution.



Standardize Interoperable Protocols

- **Prevent protocol silos** through neutral Linux Foundation (AAIF) governance.
- **Connect agents vertically** via MCP and horizontally via A2A protocols.

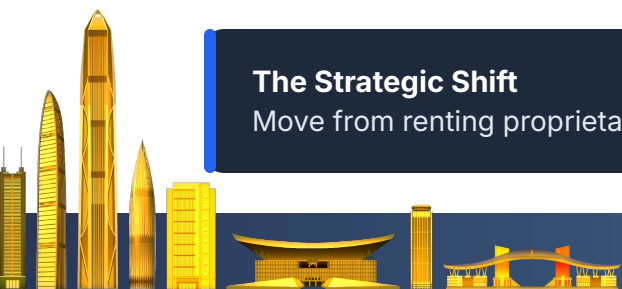


Achieve True Digital Sovereignty

- **Decouple execution** from proprietary cloud APIs and silicon lock-in.
- **Build operational autonomy** across PyTorch, CNCF, and AAIF stacks.

The Strategic Shift

Move from renting proprietary API wrappers to building a connected, open operational fabric.



Thanks!

