uracle

AURDA

Enterprise AI Operations, Perfected

2026

Challenges in AI Infrastructure Operations

What's needed now is a new operational framework capable of controlling complex infrastructure.

Limits of Traditional Infrastructure Management

The spread of generative AI has sharply increased management points across models, data, and GPU/NPU resources. Yet infrastructure operations remain service-centric, making resource control and optimization difficult for complex workloads.

53% of enterprises lack skilled personnel for high-density AI infrastructure
Flexential (2024)

Diversifying Infrastructure & the Need for Efficient Management

GPUs remain core resources, but cost and power constraints are driving adoption of dedicated inference hardware like NPUs. This mix of heterogeneous hardware makes unified resource management and LLM serving optimization difficult, leaving costly resources idle or underutilized.

AI infrastructure utilization averages just 50–70%, with the rest sitting idle
The State of AI Infrastructure at Scale (2024)

Rising Complexity in Enterprise Environments

Existing platforms offer authentication and security, but governance stays fragmented by service, making integration difficult. As AI spreads, management points—data access, model calls, API integration—multiply, making enterprise-grade unified authentication, auditing, and security essential.

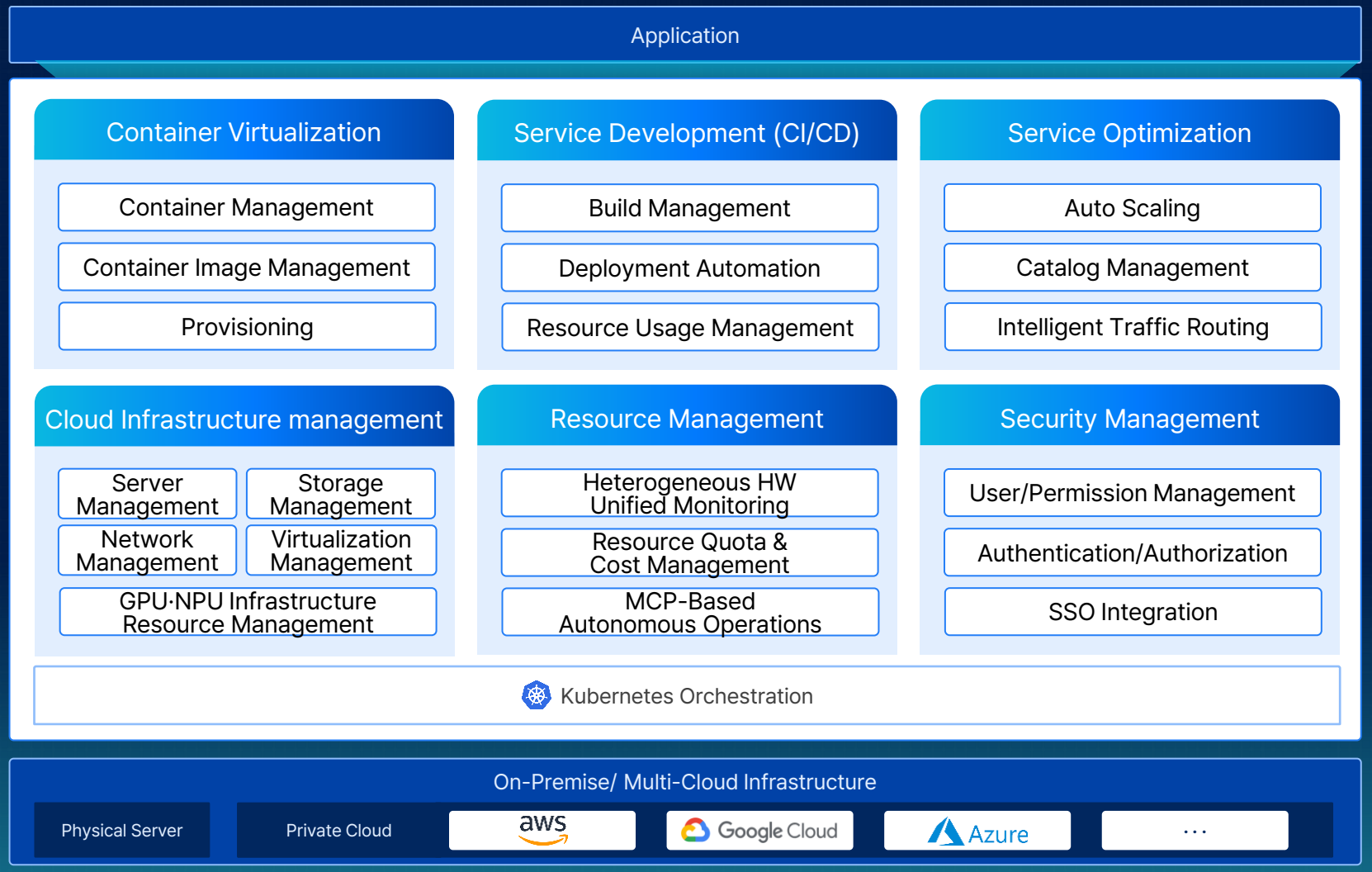
Running AI workloads directly on PaaS raises operational complexity by 2x on average
Nutanix (2023)

Core Technologies for AI Infrastructure Operations

A new operational framework requires organic integration of resources, deployment, visibility, and security.

Category	Core Function	Key Technologies	Expected Effects
Unified Resource Management	Cloud-neutral abstraction of heterogeneous hardware—GPU, NPU, CPU—for efficient compute resource utilization and auto-scaling	GPU/NPU Operator Resource Scheduling Distributed Storage Service Heterogeneous HW Pooling	**Maximized Resource Utilization** **Reduced Operating Costs**
Deployment & Automation	Unified pipeline automating the entire process from code changes to deployment	GitOps Auto-Deployment Intelligent Traffic Routing Image Registry Management All-in-One Serving Gateway	**Faster Deployment** **Zero-Downtime Operations**
Operational Visibility & Recovery	Unified collection of metrics, logs, traces, HW physical indicators (power), and LLM inference metrics (TTFT/TPS) for operational visibility	Metric/Log Collection Trace Analysis HW Physical Monitoring LLM Performance Tracking	**Proactive Failure Response** **Improved Operational Efficiency**
Intelligent Autonomous Operations	MCP (Model Context Protocol)-based AI agents that interpret natural-language commands to autonomously perform deployment, scaling, restarts, and other infrastructure tasks	MCP Standard Protocal Infrastructure Control Agent Natural Language Processing (NLU) Autonomous Orchestration	**Minimized Operational Complexity** **Task Automation**
Authentication & Governance	Unified security framework spanning authentication, authorization, secrets management, and audit logging across all services	Unified IAM Policy-Based Access Control Audit Logging & Tracking Security Policy Management	**Strengthened Security** **Regulatory Compliance**

AURDA: Defining the Standard for AI Infrastructure Operations



Operational Efficiency

Automated resource management and deployment minimize operational burden while maximizing GPU/NPU/storage utilization.

Visibility & Optimization

Unified management of metrics, logs, and traces enables failure prediction and gives you real-time visibility into your infrastructure at a glance.

Enterprise Security

Standardized authentication, authorization, and audit logging ensure service stability and regulatory compliance across the organization.

Scalable Infrastructure

Flexible scalability spanning multi-cloud and on-premise environments allows agile response to enterprise growth and change.

AURDA, AI Infrastructure Management Solution

From heterogeneous hardware to AI agent integration—one platform, delivering high efficiency and zero downtime.

Heterogeneous Hardware Abstraction

No Vendor Lock-In | Unified Control | Lower Cost

All-in-One Serving & Traffic Control

One-Click Deployment

Zero-Downtime Failover

Intelligent Routing & A/B Testing

Easy to Use | Zero Downtime | Simple Control

AURDA
One Unified Platform

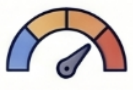
MCP-Based Agentic ChatOps


Allocate 2 A100 GPUs to the AI Team namespace.


Done — 2 A100 GPUs allocated.
(Running allocate_gpu)

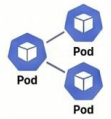
Natural-Language Requests | Operational Ease

End-to-End Unified Monitoring



GPU/NPU Temp

TPS (tokens/sec)



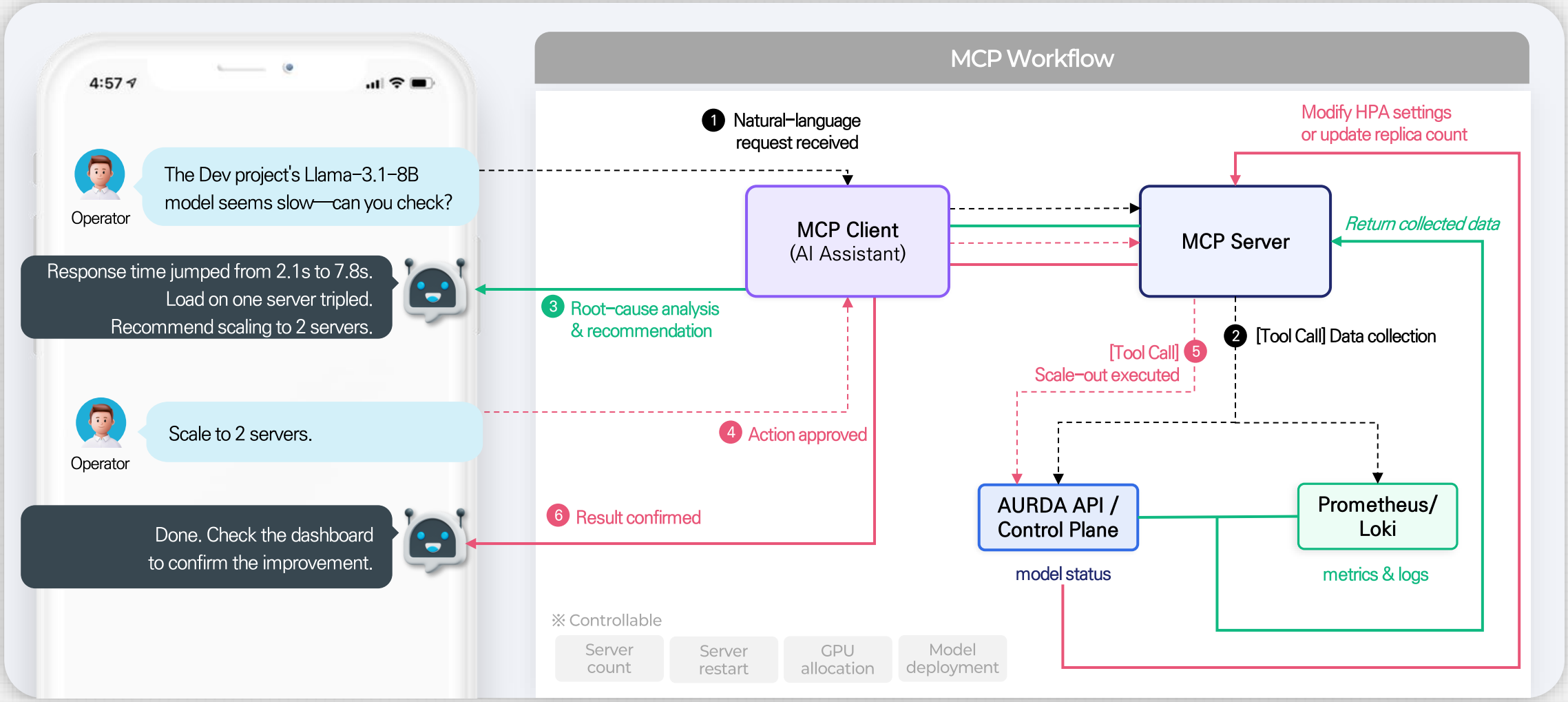
Container/Node Management

Security Audit Trail

Single-Dashboard Tracking | Fast Root-Cause Analysis

Strengths: MCP-Based Agentic Infrastructure Control

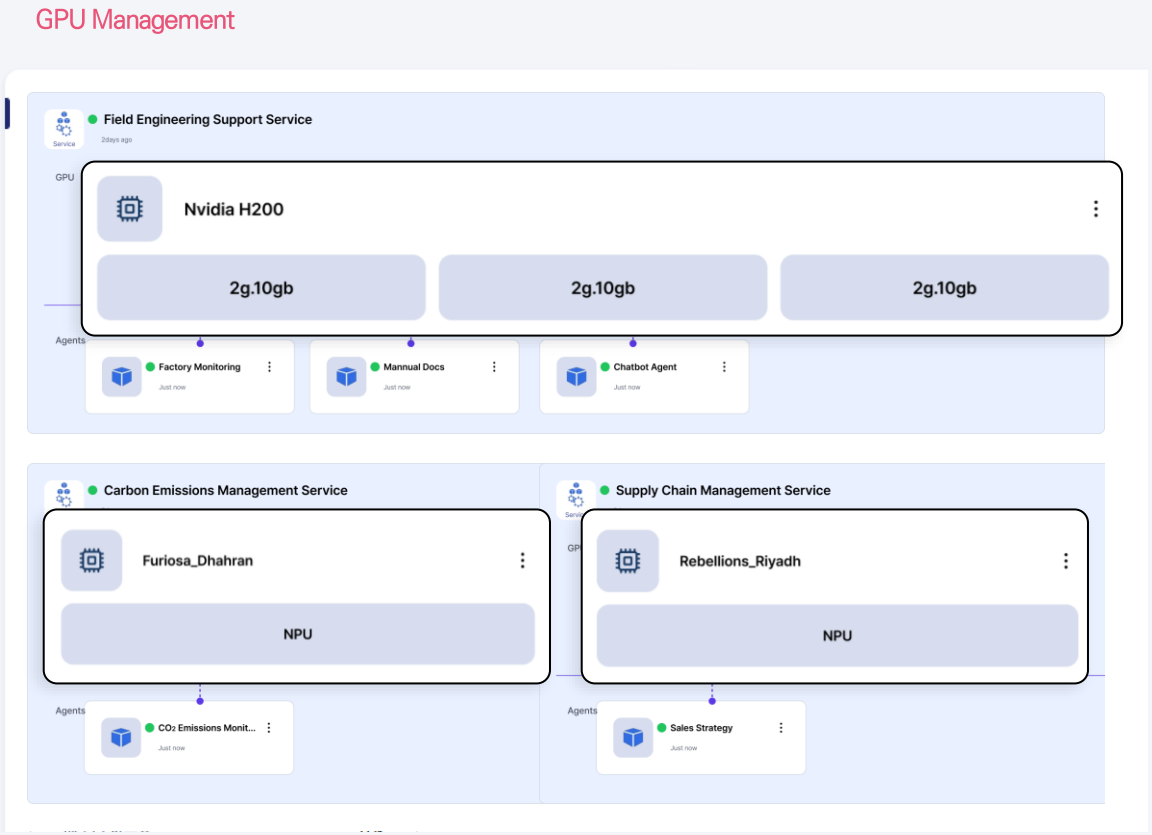
Natural-language control cuts incident response time and automates routine ops.



Strengths: Unified Heterogeneous HW Monitoring (GPU+NPU)

Manage NVIDIA GPUs and domestic NPUs on one platform—optimal AI infrastructure, no vendor lock-in.

GPU · NPU Unified Dashboard



Business Impact

No Vendor Lock-In

Freely mix GPUs and NPUs without being tied to one cloud or vendor.

Lower Costs

Cut power use by 35–70% using NPUs for inference workloads at equal load.

Single-Console Control

New hardware is instantly recognized and managed from the same UI.

Zero-Downtime Scale-Out

New GPU/NPU nodes auto-register—scale instantly, with no service interruption.

Key Features : Role-Based Operations Flow

From AI infrastructure setup to agent operations—all in AURDA.

Infrastructure Operator
Infrastructure & Foundation Setup

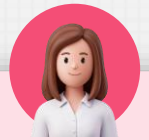
Cluster Setup	· K8s node registration · air-gapped deployment
Resource Allocation	· GPU/NPU pooling · MIG isolation
GitOps Deployment	· ArgoCD/Helm · config management
Security Setup	· IAM/RBAC · Namespace isolation

A securely isolated AI infrastructure

AI Platform Admin
AI Runtime Configuration

Model Serving	· vLLM setup · inference engine deployment
Traffic Control	· routing/failover · A/B testing
Inference Optimization	· NPU utilization · MIG templates
Observability	· LLM tracing · TTFT/TPS collection

A production-ready AI service runtime



AI Developer
Build & Run AI Services

RAG Pipeline	· vector DB integration · document embedding
AI Agent Creation	· LLM model integration · workflow design
Service Deployment	· group-based permissions · endpoint issuance
Guardrails & Monitoring	· safety policies · response quality tracking

AI services, built and running

Key Features : Unified GPU Resource Management & Allocation

Manage diverse GPU resources centrally and allocate them precisely by workload—cutting waste and maximizing operational efficiency.

MIG-Based Dynamic GPU Partitioning

Split a single GPU into isolated instances with dedicated compute and memory per workload.

Automated GPU Operator Scheduling

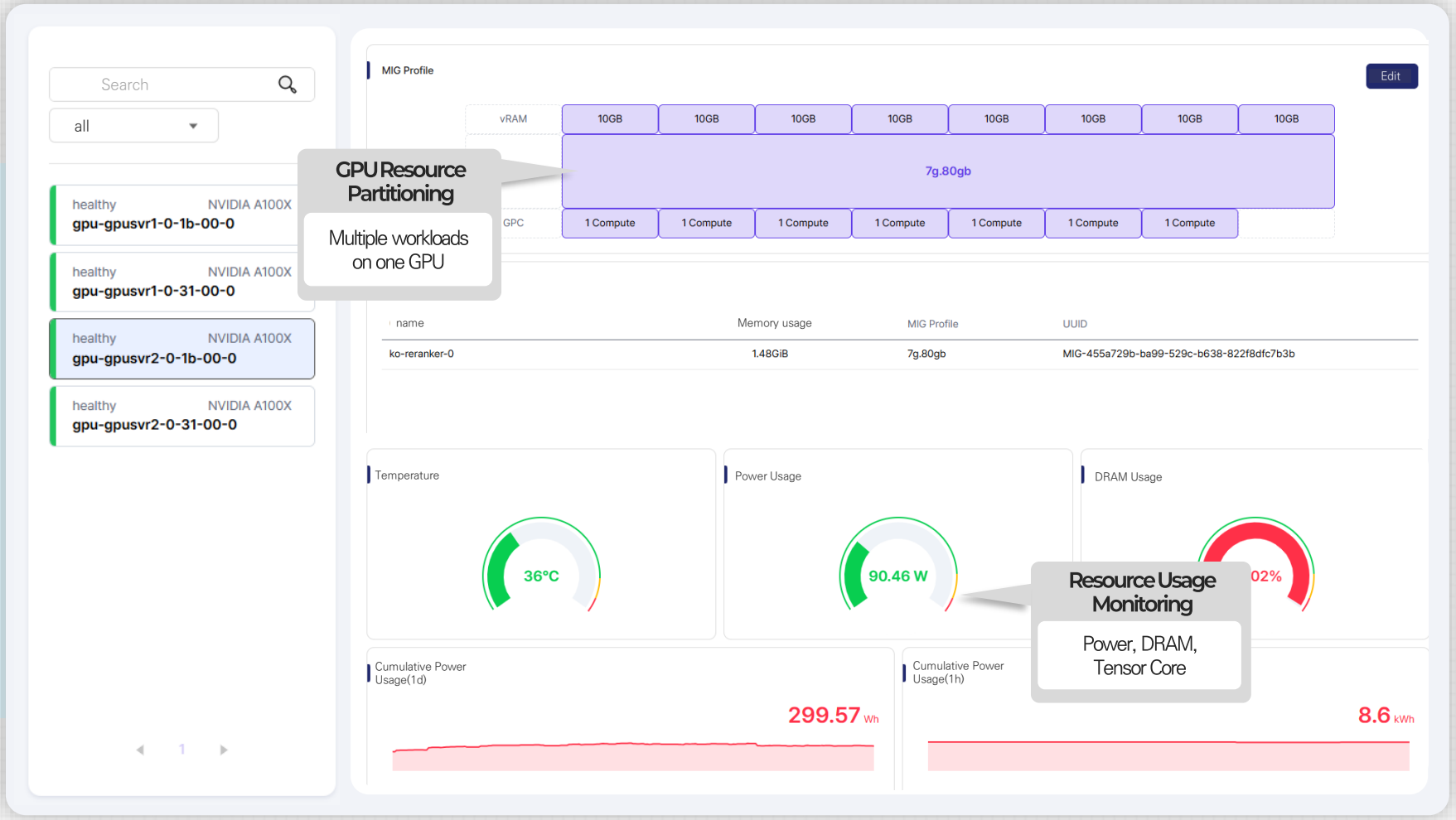
Fully automates everything from driver installation to runtime, instantly placing each workload on the optimal node.

Namespace-Level Quota Management

Set CPU/Memory/GPU limits per team or project to prevent any single workload from monopolizing shared resources.

Heterogeneous Hardware Abstraction

Run and control NVIDIA GPUs and NPUs together from a single console.



Key Features : GitOps Deployment

Manage infrastructure and app deployments as code from a single Git repo—roll back safely, anytime.

ArgoCD-Based Declarative Config Management

Auto-sync cluster state to match Git—keeping all infrastructure config consistent as code.

Helm Chart-Based Modular Deployment

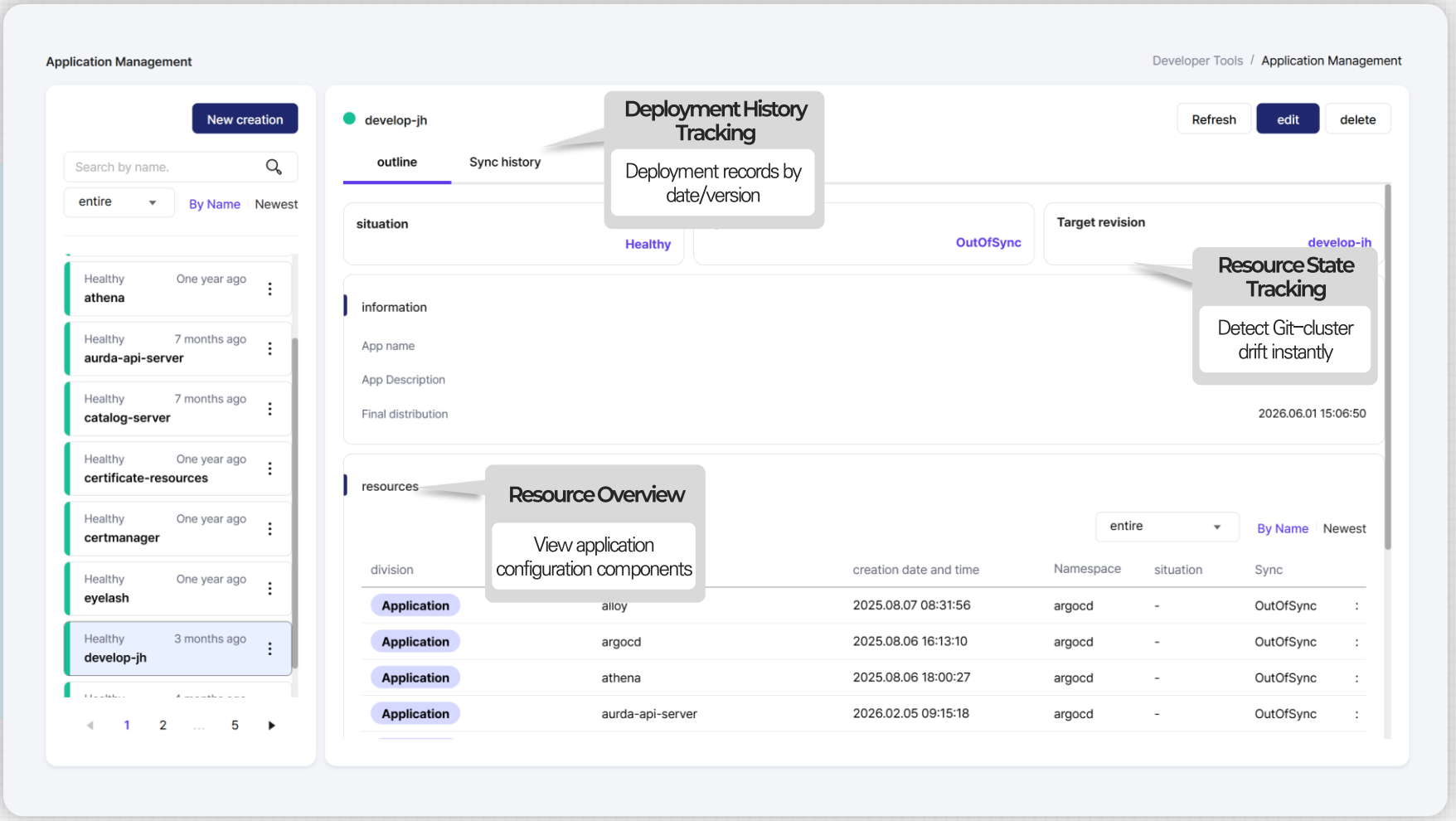
Standardize AI service and infrastructure components into packages for environment-specific deployment (Dev/Prod).

Real-Time Sync & Drift Detection

Instantly detect drift between Git and live cluster state—auto-remediate or alert operators.

Deployment History & Instant Rollback

Every change tracked transparently via Git commits—roll back to any prior version instantly on failure.



Key Features : Unified Model Management

Standardize model info, runtime environments, GPU resources, and deployment policies per model—reducing operational complexity.

Model Catalog & Version Management

Catalog commercial and open-source models, tracking metadata, versions, and status in one system.

Per-Model Inference Configuration

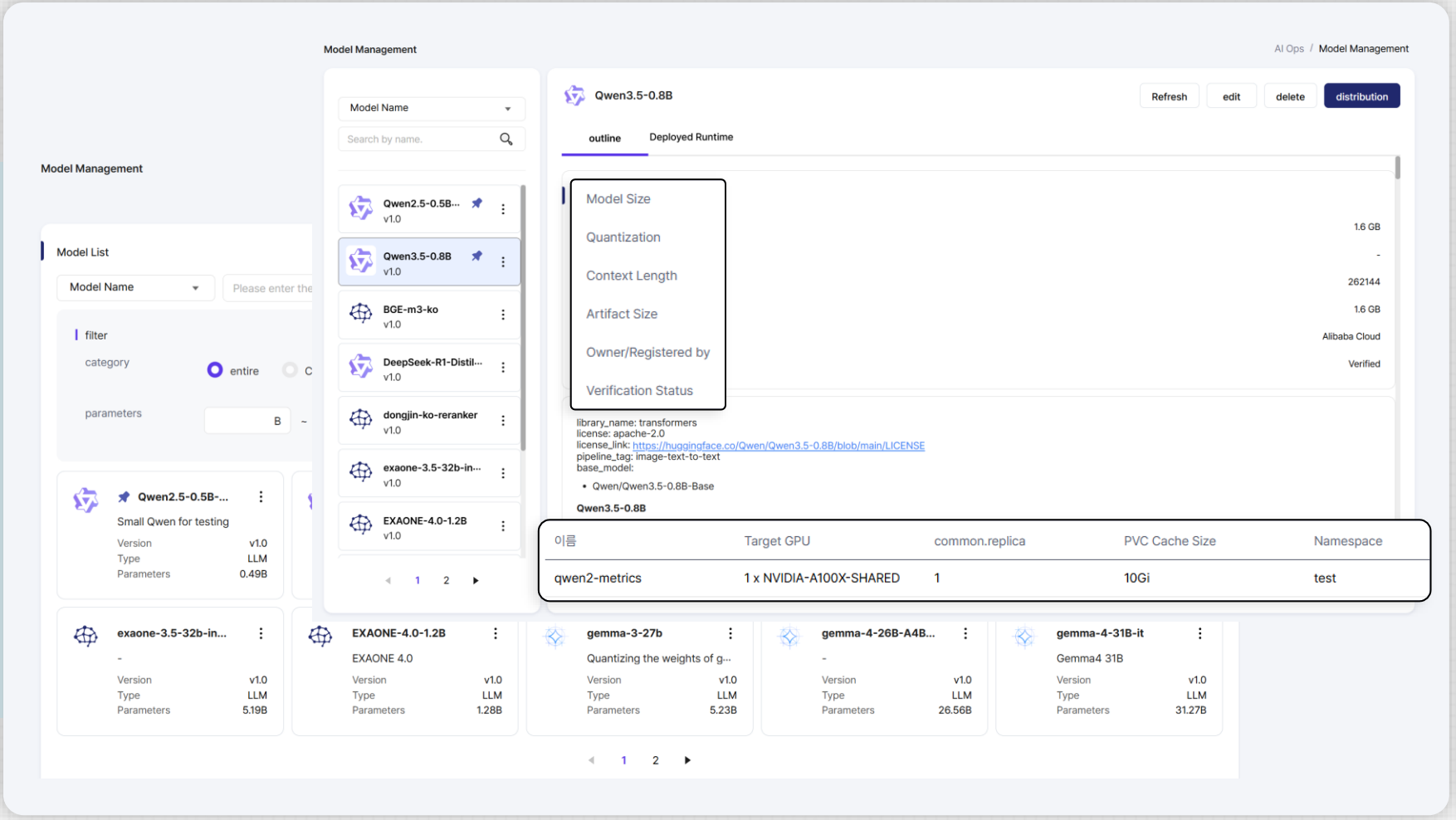
Configure serving engines like vLLM and inference settings for each model's optimal runtime.

Model Storage Management

Securely store large model files via object storage—supporting stable operation even in air-gapped environments.

Model Deployment Configuration

Configure deployment methods and replica counts to match service scale and traffic, all governed by a consistent deployment policy.



Key Features : High-Performance Inference Engine

Deploys diverse LLMs to fit your service environment, delivering stable model services through standardized APIs and high-performance inference.

UI-Based Model Deployment

Select a model and runtime environment to quickly configure and deploy model services with no complex setup

High-Performance Inference

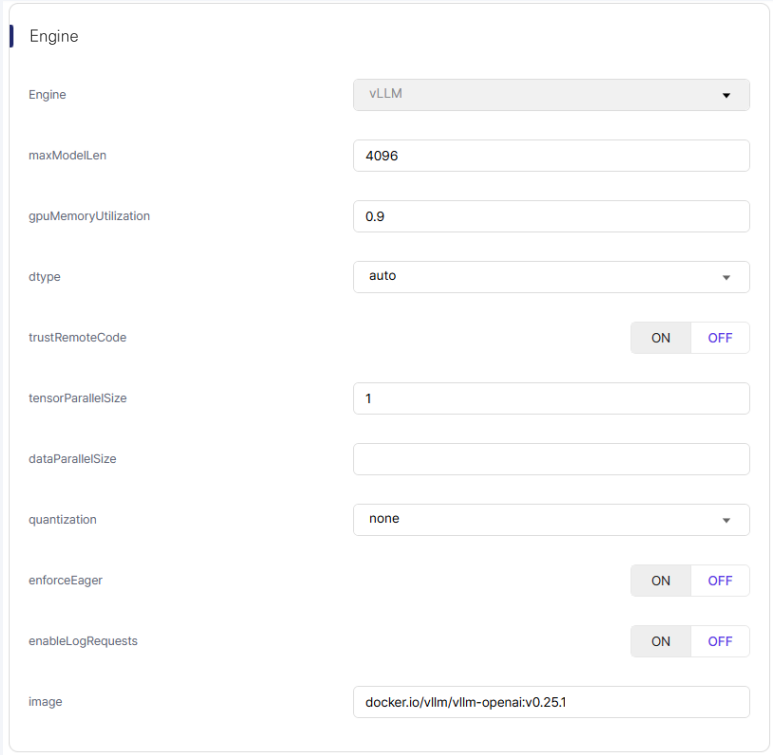
vLLM-based efficient request processing with caching improves concurrent throughput and inference performance

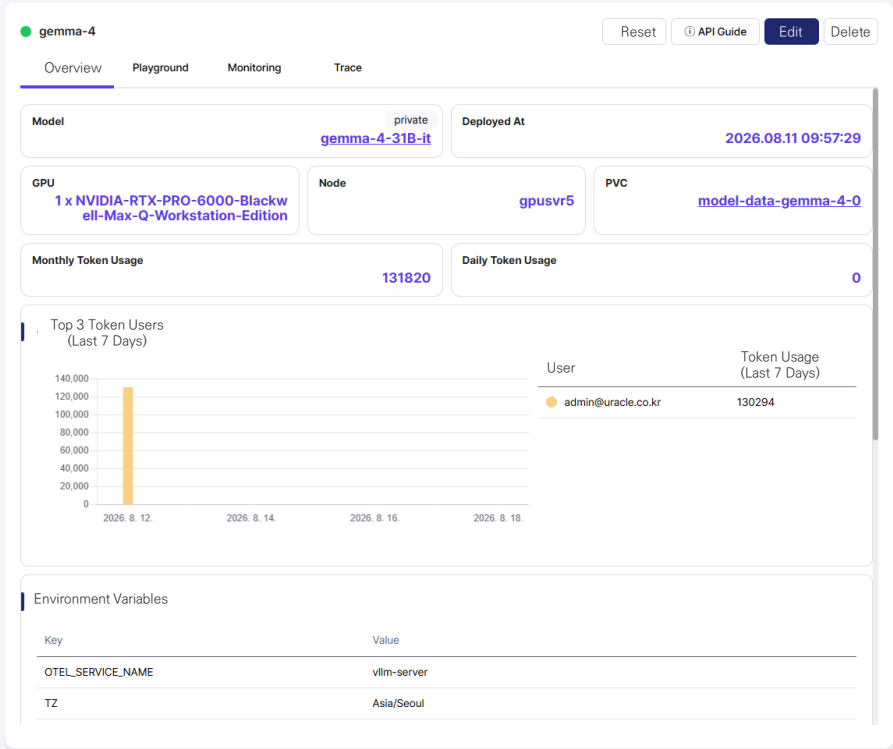
Single Endpoint per Model

Consolidates distributed model instances into a single endpoint, enabling consistent model invocation across services

Standard Inference API Support

OpenAI Compatible API makes it easy to integrate with existing AI services, supporting streaming and other invocation methods





Key Features : High-Performance Model Inference

Elastically scale inference resources across Multi-GPU and Multi-Node environments, and optimize resources for each processing stage to ensure stable inference performance for large-scale AI services.

Distributed Inference Environment

Distributes inference workloads across multiple GPUs and nodes, flexibly handling large-scale models and high request volumes.

Inference Stage Optimization

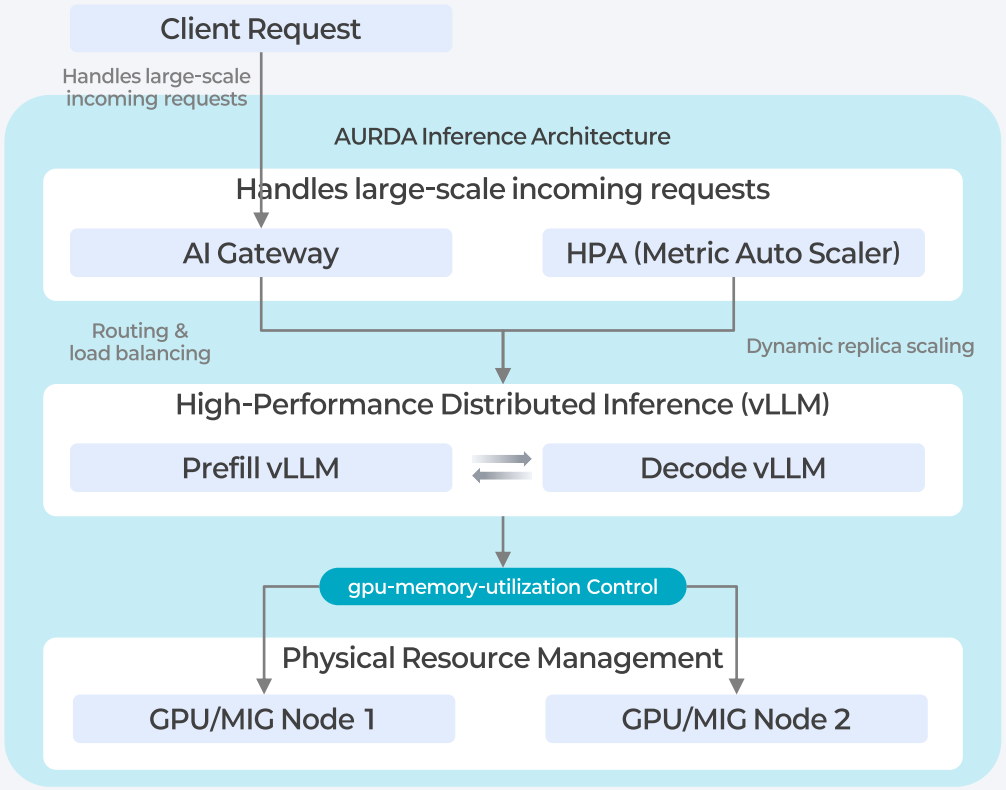
Separates Prefill and Decode processing stages, operating GPU resources efficiently based on each stage's characteristics.

Inference Compute Efficiency

Reuses KV Cache to cut redundant computation and boost inference efficiency.

Elastic Inference Resource Scaling

Independently scales GPU resources per stage to adapt to changing inference traffic.



- **AI Gateway (Request Intake Control)**
  : Supports weighted routing and automatic retry to fallback targets.
- **Auto-Scaling (Elastic Scaling)**
  : Automatically adjusts the number of vLLM instance replicas based on HPA metrics.
- **vLLM Serving Engine (Performance Optimization)**
  : Applies TP (Tensor Parallelism) / PP (Pipeline Parallelism) and operates based on gpu-memory-utilization.
- **GPU Operator & MIG (Resource Distribution)**
  : Allocates GPU resources per namespace and manages dynamic profiles.

Key Features : Inference Traffic Control

Controls traffic based on inference request volumes and service status, ensuring stable model serving even in the event of failures or overload.

Traffic Protection & Failover

Uses policy–based rate control with Circuit Breaker and Fallback to keep services stable under overload or failure.

Load Balancing

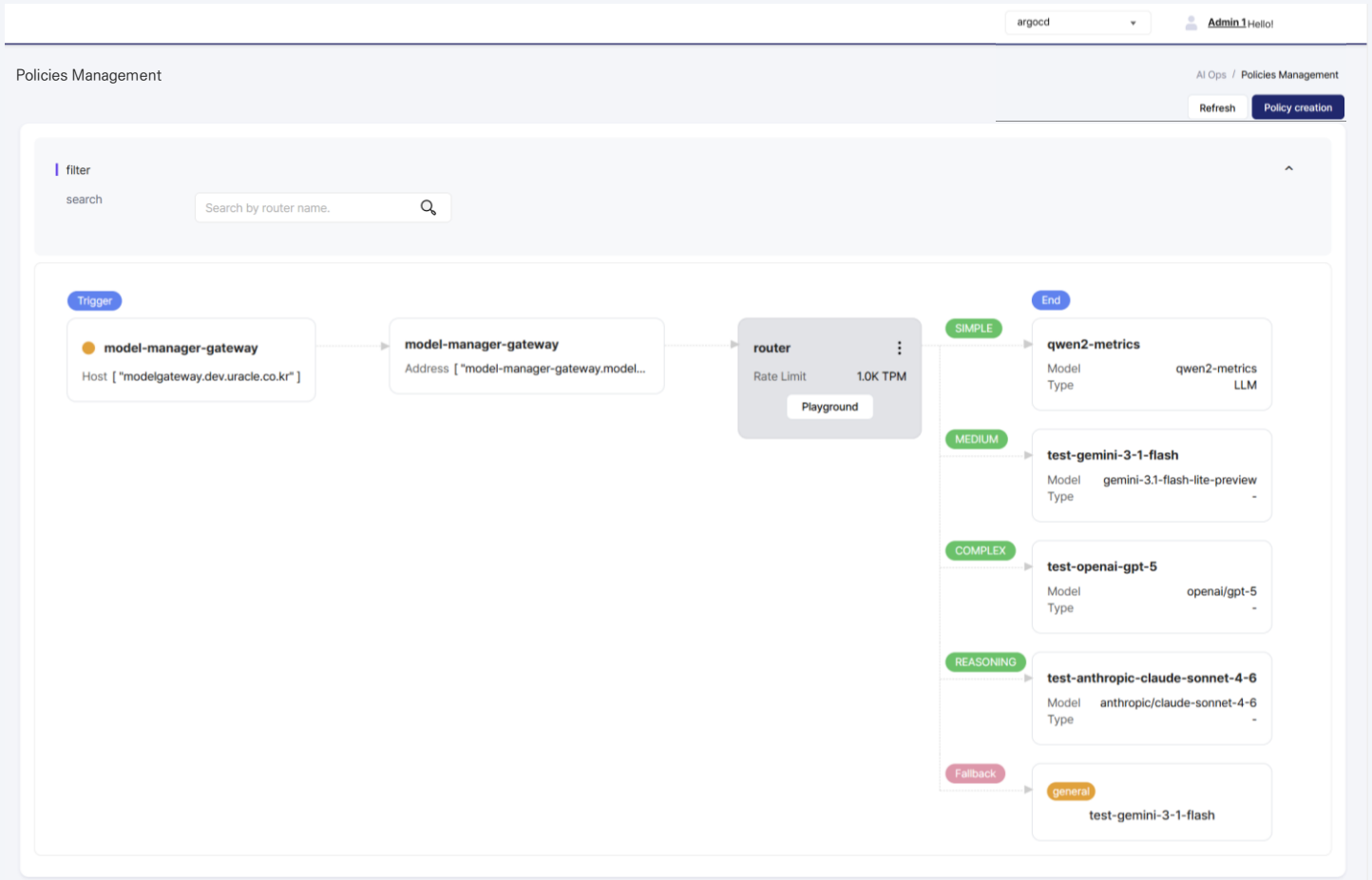
Auto–distributes inference traffic based on real–time load, latency, and priority.

Request Rate Policy Management

Controls per–service traffic limits via RPM, TPM, and concurrent requests.

Priority–Based Request Handling

Sets processing priority based on service importance and policy.



※ RPM (Requests Per Minute)
TPM (Tokens Per Minute)

Key Features : Advanced Inference Routing Strategy

Selects the optimal model and processing path based on cache state, query semantics, and context to maximize inference efficiency.

Cache-Based Routing

Distributes requests based on KV/Prefix Cache state to cut redundant computation.

Semantic Model Selection

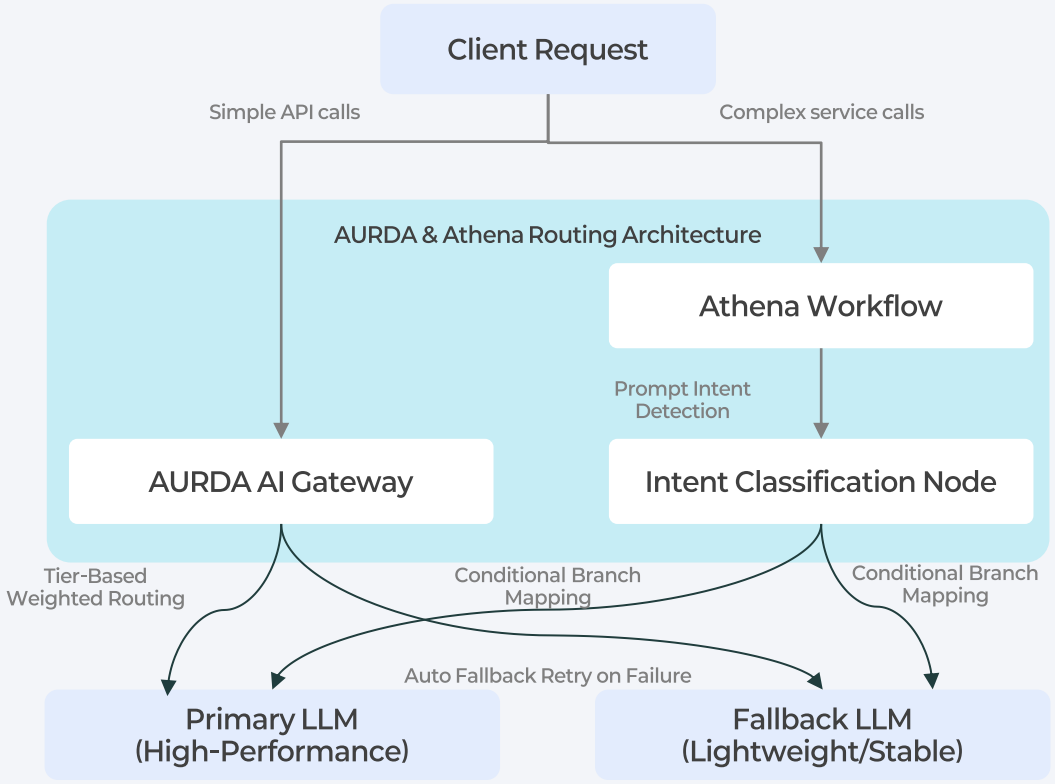
Analyzes query intent and domain to auto-route to the best-fit model.

Context-Based Branching

Routes requests by context size and request conditions to the right model/endpoint.

Service-Level Routing Policy

Auto-routes traffic to dedicated models/endpoints per predefined service rules.



- **AI Gateway-Level Traffic Control**
  : Namespace-level policy isolation and access control via a LiteLLM-based OpenAI-compatible API endpoint.
- **Zero-Downtime Failover Routing**
  : Tier-based weighted distribution with automatic fallback retry on model/node failure.
- **Intent-Based Workflow Routing**
  : Maps LLMs to business scenarios using Langflow's intent classification and conditional branching.

Key Features : Inference Operations Visibility

Tracks the full pipeline in real time—from request to inference engine to GPU resources—supporting stable AI service operations through performance, usage, and error insights.

Inference Performance Monitoring

Tracks per-model performance and response status in real time via RPS, Latency, TTFT, TPOT.

Inference Engine Health Analysis

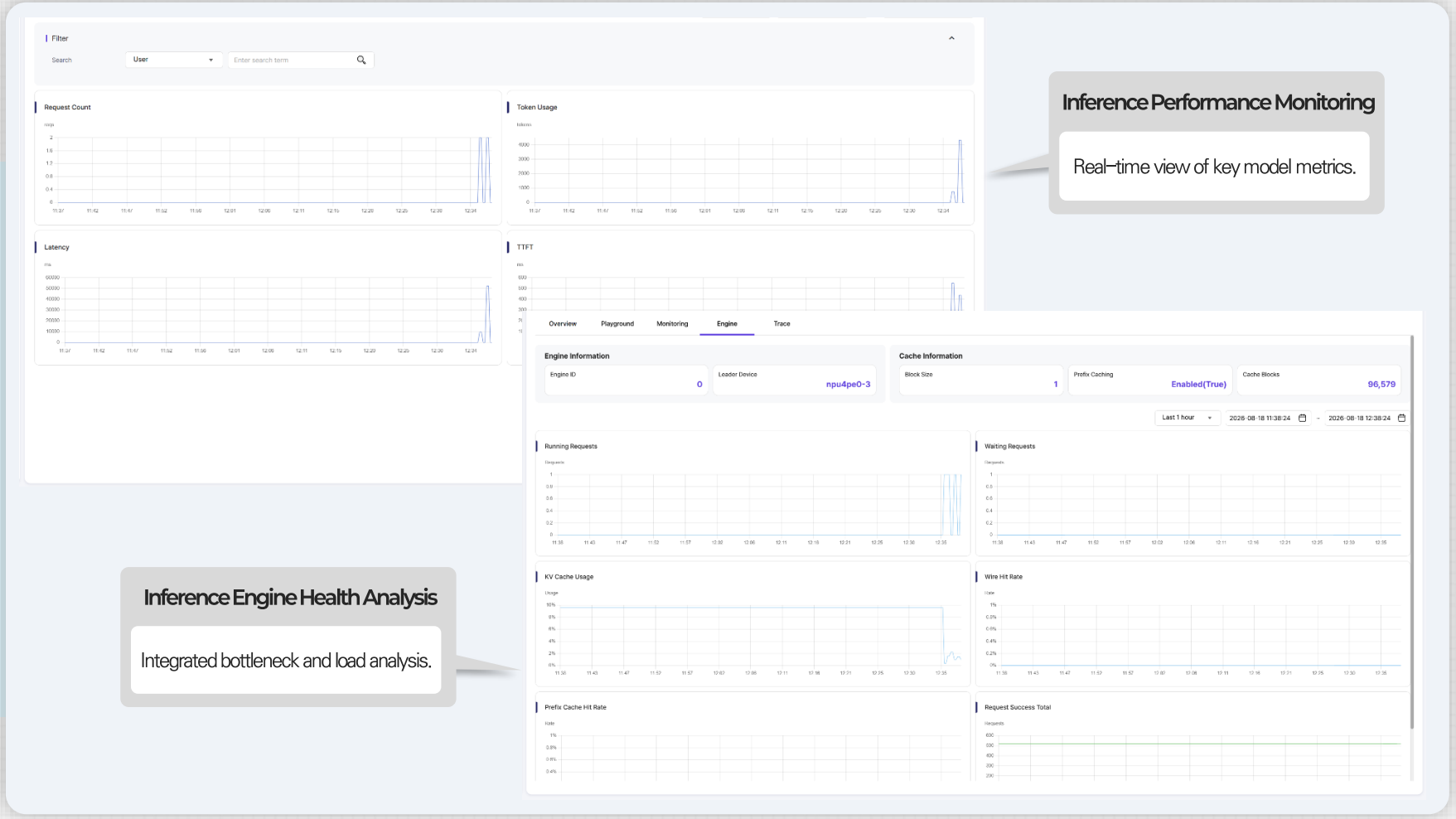
Integrates queue, cache, and GPU status to analyze engine load and bottlenecks.

End-to-End Request Tracing

Traces the full path from Gateway to Routing to Inference in one view to pinpoint latency and errors.

Operation History & Audit Management

Logs API key usage and policy changes for traceability and audit support.



Key Features : AI Service Monitoring

Tracks the entire pipeline in real time, from infrastructure resources to LLM inference, enabling fast response to service anomalies through logs and alerts.

Full-Pipeline LLM Inference Tracing

Traces each step from request to response to instantly identify bottlenecks and delays.

Unified Metrics Monitoring

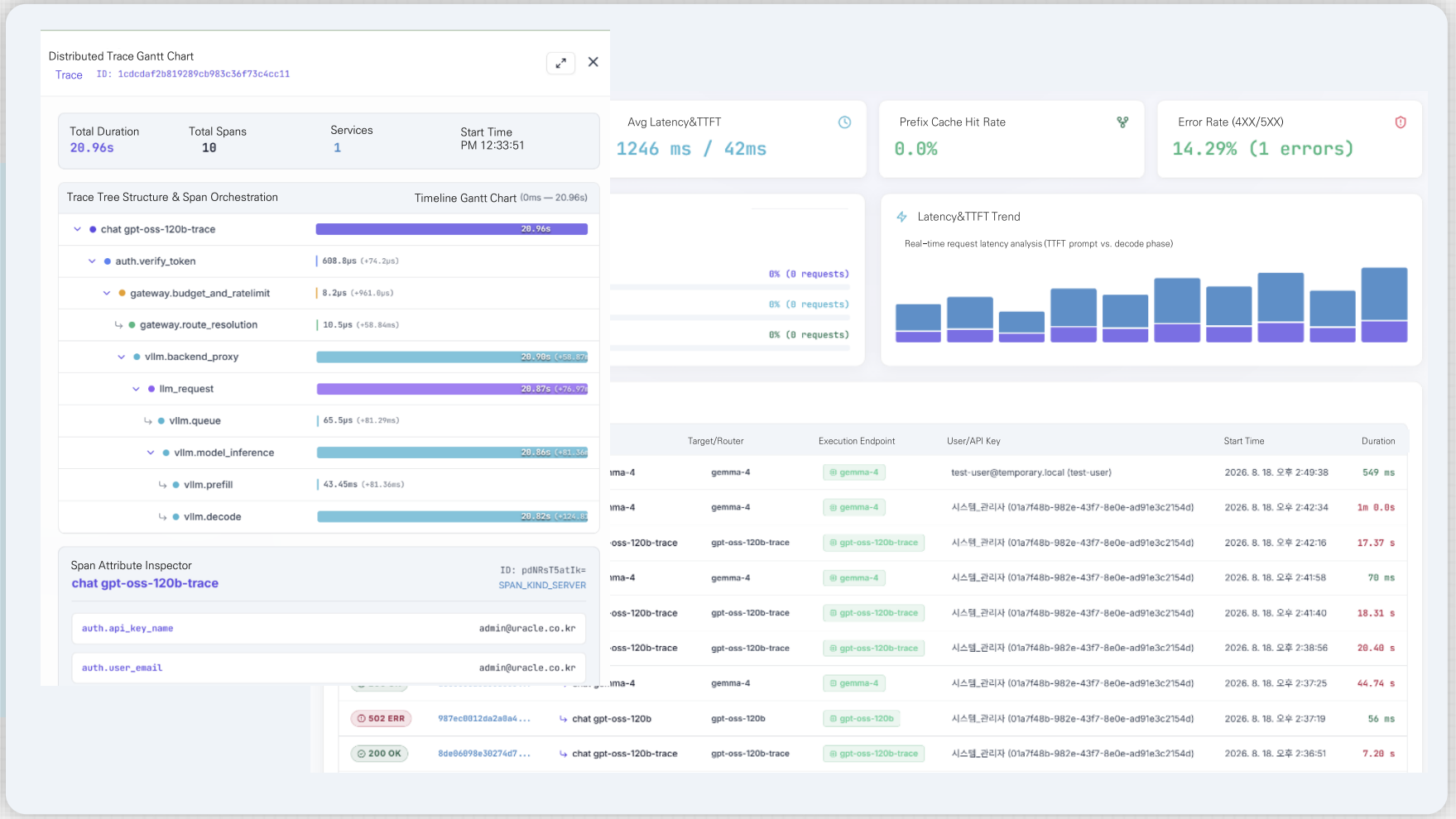
Tracks GPU and LLM performance metrics on one dashboard for integrated service health management.

Centralized Log Management

Aggregates logs across all components for fast troubleshooting during incidents.

Threshold-Based Real-Time Alerts

Detects anomalies in GPU status, error rates, and key metrics, using alert history to support rapid response.



Key Features : AI Usage & Cost Management

Analyzes AI usage and cost across models, users, and organizations, enabling efficient resource management through budget policy.

User/Org Usage Management

Aggregates model calls and token usage by user and team to track AI service adoption across the organization.

Per-Model Usage Analysis

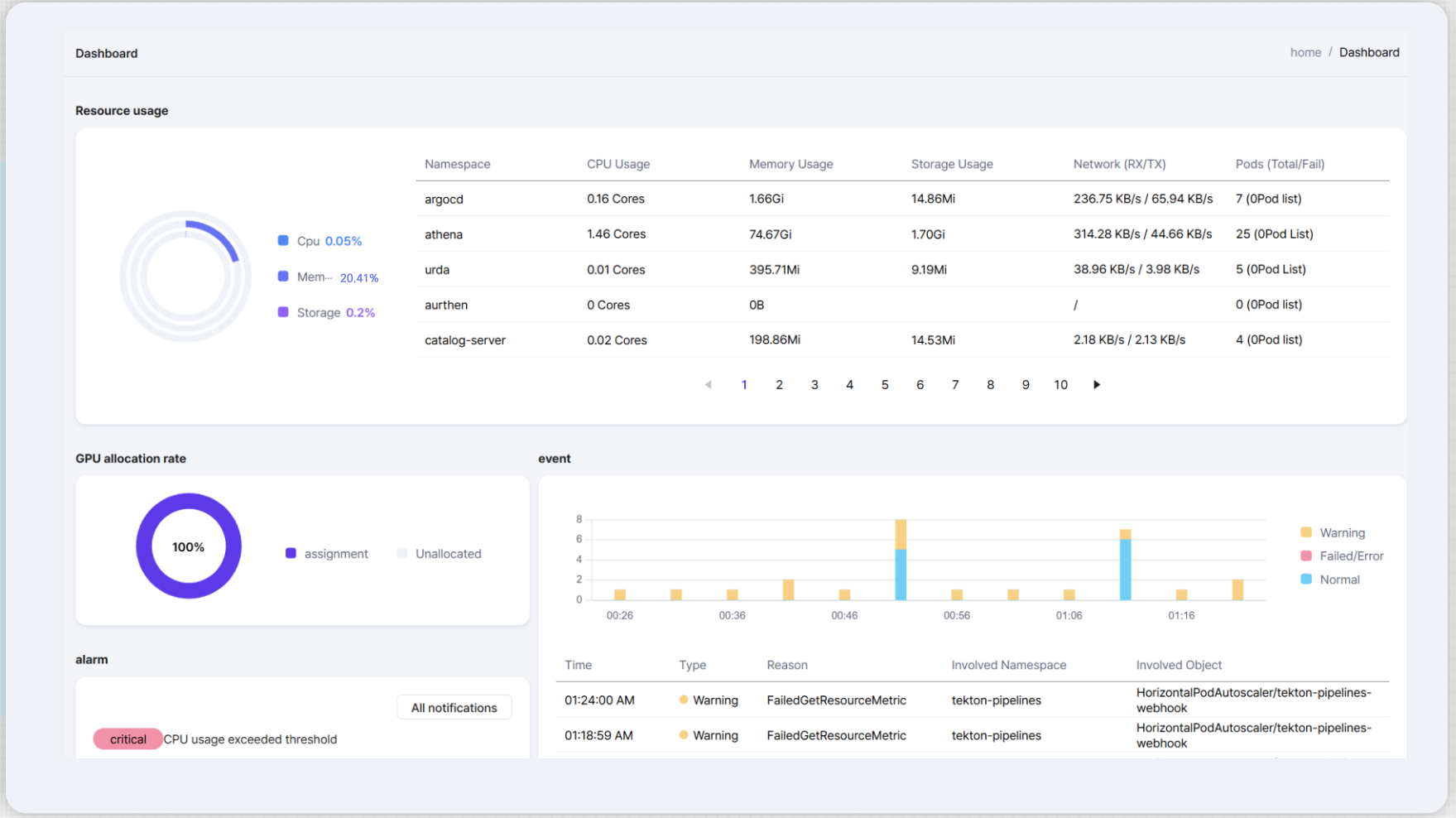
Analyzes request and input/output token usage per model to compare activity across services and models.

AI Cost Tracking

Calculates cost per model based on token usage, with unified cost tracking by service and organization.

Budget & Usage Limit Management

Manages budget policy and usage by org and service to keep AI resource use within budget.



Key Features : Security & Access Control

Centrally manages access permissions and service policies at the user, organization, and API level to keep AI models and infrastructure secure.

Role-Based Access Control (RBAC)

Manages granular access to models and resources based on user and organizational roles.

Enterprise Account & API Authentication

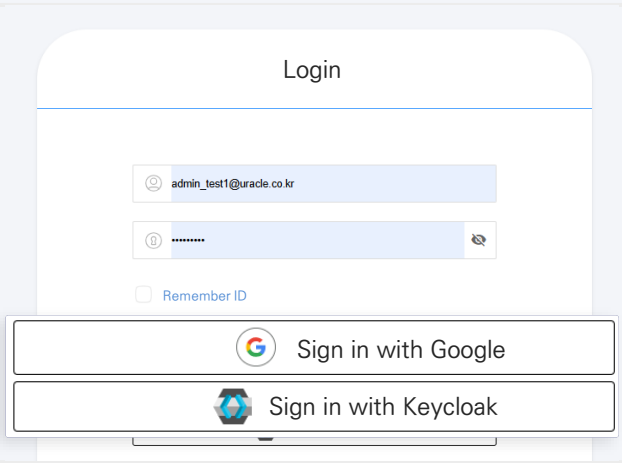
Integrates with SSO, external accounts, and API key auth for seamless enterprise identity management.

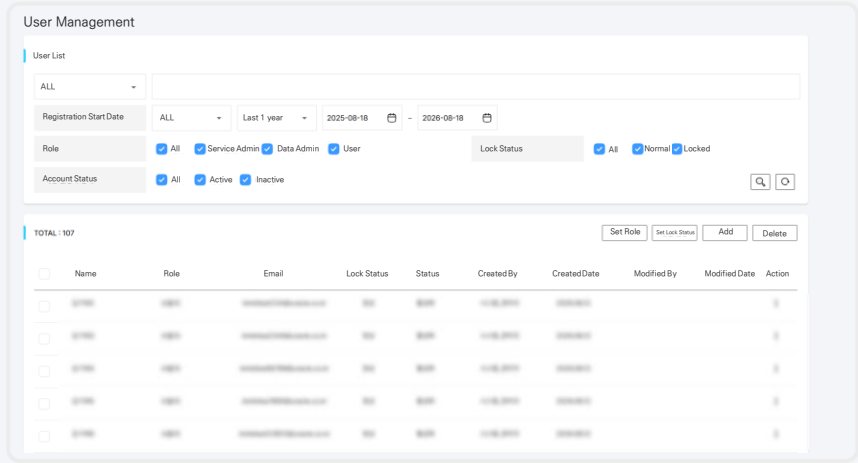
Model Access & Usage Policy Management

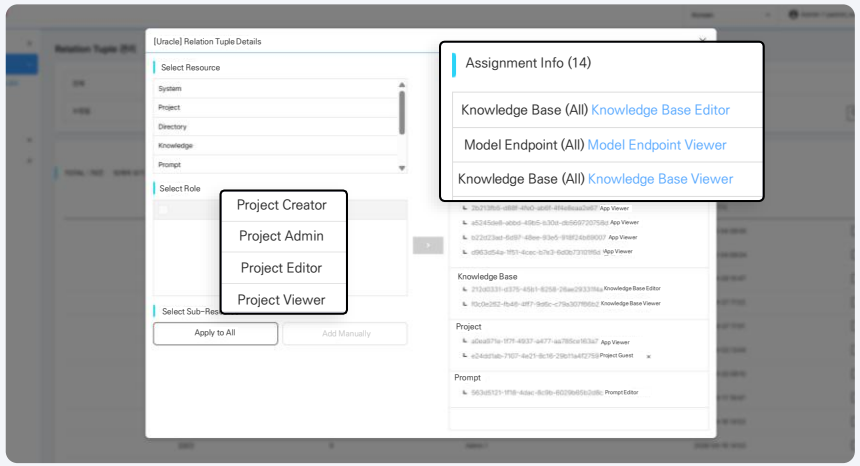
Sets model access permissions and request limits per user/team to govern AI resource usage by policy.

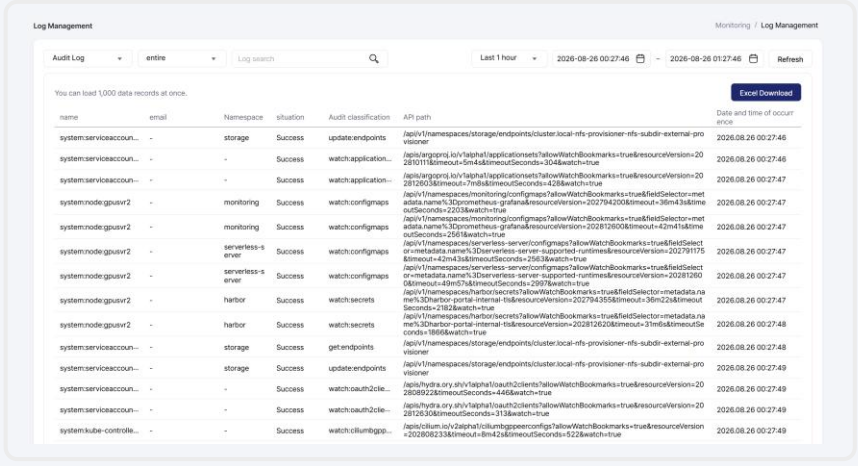
Security Event & Change History

Logs key events such as logins, permission changes, and policy updates to support operational tracking and security audits.





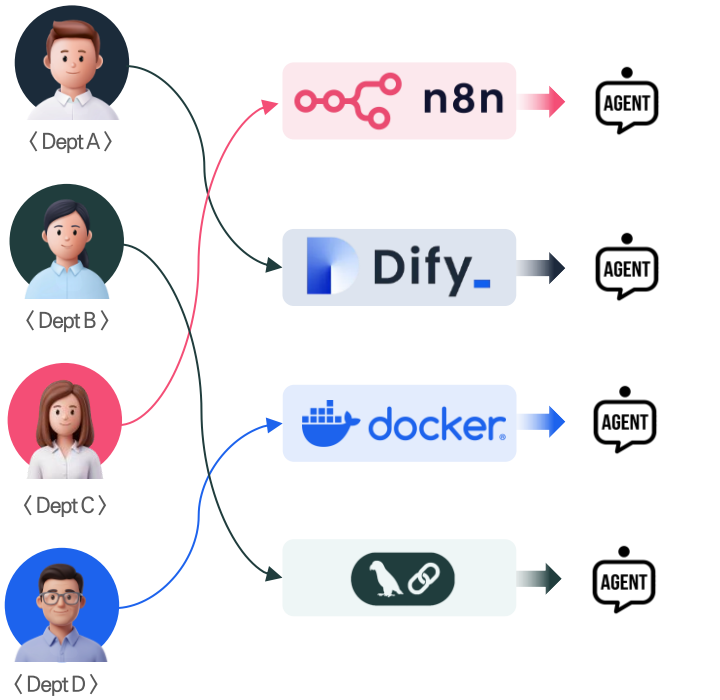




Our Roadmap : Universal AI Runtime

Run and control heterogeneous AI services—N8N, Docker, legacy agents—on a single Athena · AURDA environment.

Siloed Heterogeneous AI Environments

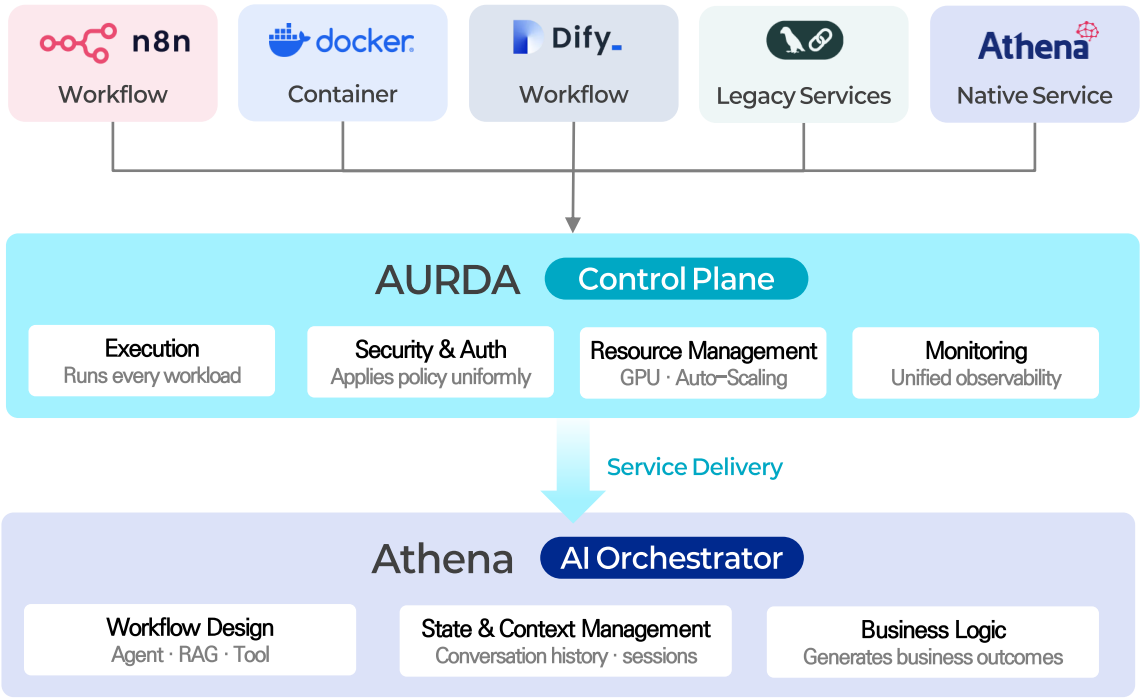


Method · Security policy — all different

Independent operation per service means inconsistent security, resource management, and monitoring standards

—making org-wide AI governance impossible.

Unified Operations via Control Plane



AURDA's Control Plane handles execution, security, and resources under one standard —while Athena orchestrates business logic.

uracle

Thank you

H. uracle.co.kr

E. sales@uracle.co.kr

T. +82-2-3479-4400