Wallaroo.AI

# THE UNIVERSAL AI INFERENCE PLATFORM

www.wallaroo.ai

# Table of Contents

# 1.Overview

Going from proof-of-concept to viable AI in production is hard. This is why most AI initiatives fail. As a result of this failure, businesses risk falling behind, stunting innovation, and losing competitive advantage.

## 1.1. AI Production Deployment Challenges

- Heavy engineering effort / specialized skills needed
- Complexity and cost of production (inference) infrastructure
- Poor monitoring, troubleshooting means failure to generate value
- Inference workloads are exploding - computer vision and GenAI
- Data is spread across disparate systems leading to lack of consolidated insights capability

Wallaroo enables operationalizing AI models to any cloud, edge or on-premises environments and on diverse hardware architectures.

Providing intuitive capabilities for AI teams to deploy, manage and monitor models in production at scale with ultimate flexibility, ease and repeatability.

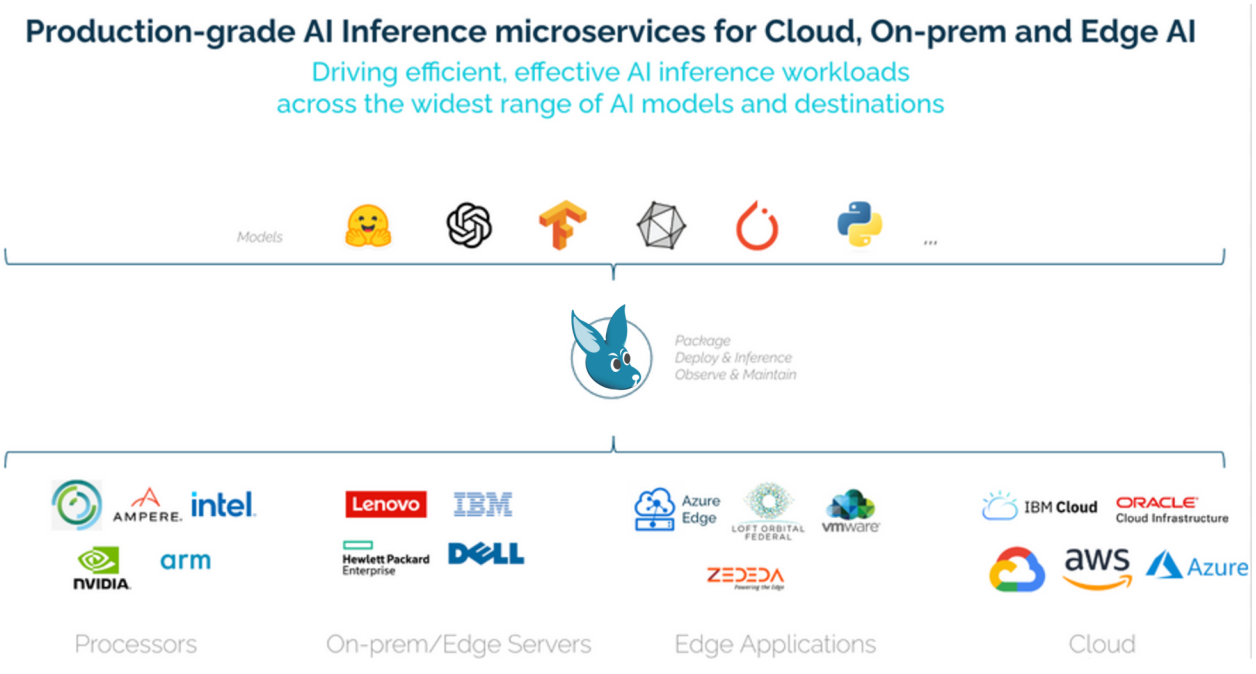
# 1.2. Where does Wallaroo fit in the AI/ML lifecycle?

The Wallaroo universal AI inference platform enables realtime and batch AI inference on any hardware type (CPU, GPU), and across various AI applications (Time Series, Computer Vision, Classification, Regression and LLMs) in cloud, edge or on-premises environments.

**Wallaroo is uniquely designed to help AI teams get models into production faster and remain focused on driving value from AI investments within the enterprise.**

**Wallaroo:**

- Automates the process of packaging and deploying production-grade AI inference microservices for any class of AI models.
- Is pre-optimized for a wide range of hardware platforms, allowing AI teams to easily target the right hardware (X86, ARM, Ampere, IBM Power10, NVIDIA etc. ) for their application;
- Supports a wide range of deployment environments, from cloud to on-prem to edge, enabling the team to deploy to wherever their applications are required to run.



# 2.Technology

Wallaroo abstracts away every infrastructure complexity and adds automation that enables AI developers to turn models into robust, production-grade inference microservice with as little effort and complexity as possible, while maintaining value with a tight feedback loop from AI model to business value. This process consists of three main steps: packaging, deployment, and observability.

## Wallaroo Helps Teams Scale

Automation through three phases of deployment

| | |
|---|---|
| **Package**<br>**Turn model into microservice.**<br>Collect dependencies.<br>Build image.<br>Test, debug, optimize, repeat | **Auto-Packaging**<br>Automatically package & **optimize.**<br>No container building.<br>Broad built-in model support. |
| **Deploy**<br>**Serve inference in cloud or at edge.**<br>Publish & configure Docker, Kubernetes.<br>Setup networking, load-balancing.<br>Hardware-specific drivers. runtimes. | **Automated, Fast Deployment**<br>Easily deploy to cloud or edge.<br>Pre-optimized for ×86, Arm, GPU, ...<br>Scales from thin edge to clusters. |
| **Observe & Optimize**<br>**Monitor drift wherever the model is.**<br>Collect & aggregate data.<br>Establish baselines.<br>Run regularly, set up alerting. | **Advanced Observability**<br>Integrated observability, **cloud and edge**<br>Automated, scheduled drift analysis<br>Built-in alerting & exploration |

Wallaroo.AI

# 2.1 Package

Wallaroo's auto-packaging capability allows an AI team to go from a trained model – anything such as a linear regression model, a CV model, model ensembles or a multi-billion parameter LLM – to a deployable service with as little as one simple command in Wallaroo. Wallaroo automatically assembles and orchestrates the model artifacts to run most efficiently on the target hardware.

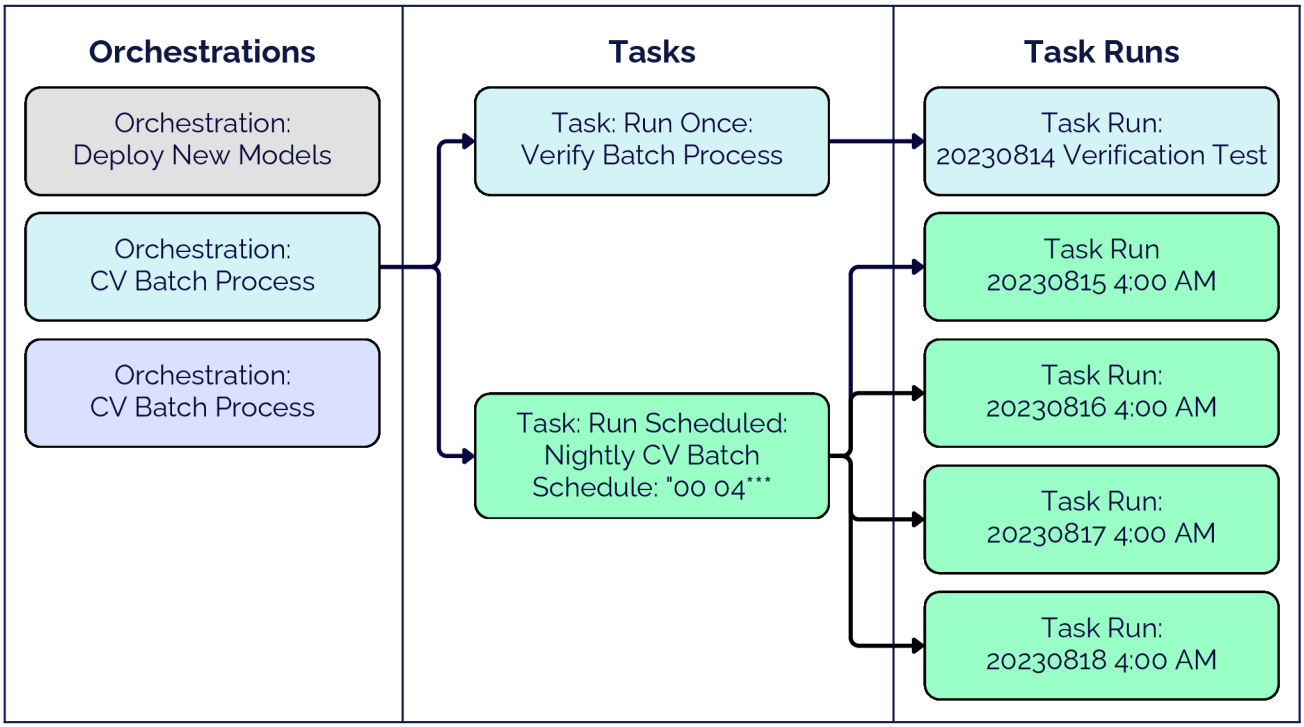
Wallaroo's Automated AI Model Packaging ensures that the AI models are consistent, portable, scalable, and reproducible across different platforms and environments helping to achieve self-service model deployment and serving in minutes with low code/no code capabilities. This means that the AI team doesn't need to spend time/effort finding and installing the correct runtimes and drivers for each platform to meet desired deployment and performance requirements.

For example, to upload an ONNX computer vision model on an x86 processor with an Intel GPU:

```
model = wl.upload_model("yolov8", "./models/yolov8n.onnx",
Framework.ONNX, arch = Architecture.x86, accel =
Acceleration.OpenVino)
```

Additionally, Wallaroo's AI Workload Orchestration allows teams to automate AI workflows tying together data sources, data sinks, and one or more pipelines. Common applications range from nightly forecasting jobs to processing videos as they are uploaded to a storage bucket to long-running services consuming data from event sources.

Wallaroo's AI workload Orchestration automates the packaging of the tools to help run and manage the related workflow tasks. The packaging automation works similarly to model packaging where the user provides, in Python code, the code building blocks and artifacts of their workflow, while Wallaroo does the rest, creating all of the artifacts and virtual environments on the backend, to enable AI developers to run their workflows on-demand or automatically on a schedule.




```
orchestration =
wl.upload_orchestration(path="./remote_inference/remote_infere
nce.zip")

task = orchestration.run_once(name="simpletaskdemo",
json_args={"workspace_name": workspace_name,
           "pipeline_name": pipeline_name})

scheduled_task =
orchestration.run_scheduled(name="simple_inference_schedule",
schedule="*/5 * * * *",timeout=120,json_args=
{"workspace_name":workspace_name,
"pipeline_name":pipeline_name })
```

# 2.2. Deploy

Inference pipeline deployment in Wallaroo leverages the Wallaroo Inference Engine ™, which provides a high-performance network interface and is pre-optimized for industry-leading hardware platforms. The Wallaroo Inference Engine is verified in conjunction with our hardware partners, and enables AI developers to realize the raw performance of their models with minimal overhead from the infrastructure.

Wallaroo automates the process of deploying an inference pipeline into Kubernetes or edge locations, reducing the process to a few lines of code. An AI team can easily specify the resources to allocate to the pipeline, select the hardware platform, and enable advanced features such as load-based auto-scaling.

```
dc = wallaroo.DeploymentConfigBuilder() \
    .cpus(4) \
    .replica_count(2) \
    .memory(“1Gi”) \
    .arch(Architecture.x86).accel(Acceleration.OpenVino) \
    .build()
pipe = wl.build_pipeline(“cv-pipeline") \
    .add_model_step(model) \
    .deploy(deployment_config=dc)
```

In the example above, the previously uploaded model is deployed as a service running on x86 nodes that will take advantage of an available Intel GPU. This deployment includes two replicas, each with 4 CPUs and 12GB of memory as well as load-balancing and integrated observability. All of this is accomplished without the AI team needing to have expertise in Kubernetes or optimizing for the specific hardware platform.

Publishing the pipeline for deployment to edge locations, whether a small single-board computer or a remote data center, looks similar to above. The edge publishing capabilities are designed to work seamlessly with typical deployment flows such as GitOps to simplify the handoff from the AI team to field ops teams who are in charge of orchestrating application deployments onto Edge environments.

```
dc = wallaroo.DeploymentConfigBuilder()
    .cpus(4)
    .gpus(1)
    .memory(“12Gi”)
    .arch(Architecture.x86).accel(Acceleration.OpenVino)
    .build()
pipe = wl.build_pipeline(“cv-pipeline")
    .add_model_step(model)
pipe.publish(deployment_config=dc)
```

# 2.3. Observe & Optimize

Finally, every deployment includes observability and monitoring baked in. Unlike traditional software, AI models are a function of the data they were trained on and are often asked to identify subtle patterns in those data.

As the patterns, and background patterns, shift, the efficacy of the model will change over time, a process commonly referred to as “drift”. In order to maximize the lifetime value of a model, teams need to continuously monitor deployed models and retrain them as needed.

Wallaroo provides integrated monitoring tools and, critically, automated alerting functionality to provide the team with actionable insights.

Additionally, Wallaroo offers the ability to evaluate and update models without any interruption to deployed inference pipelines with advanced capabilities involving A/B testing, shadow deployments and in-line model updates.

Observability is core to the Wallaroo Inference Engine. The engine is designed to capture the inputs to each inference request and the resulting outputs as observability logs. The observability logs processing is out-of-band from the inference processing to avoid adding latency to the response.

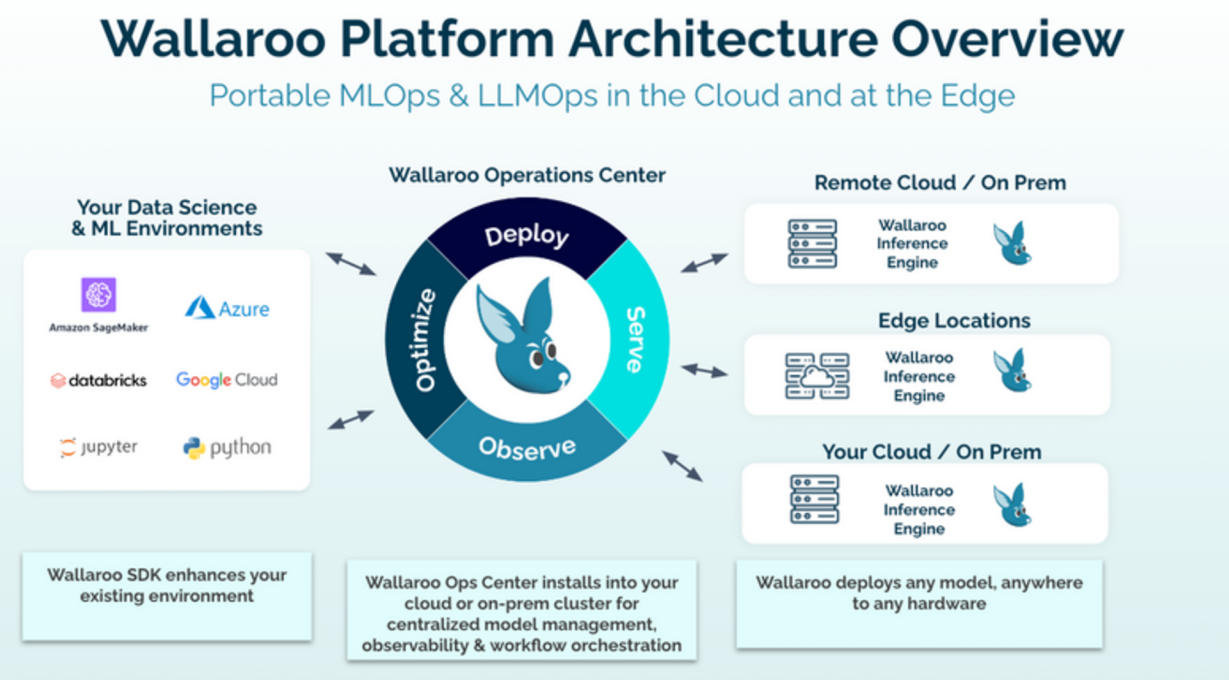
These logs are transmitted to our high-throughput log ingest system to be stored for later analysis. This observability capability is available both in cloud deployments and optionally in edge deployments when bandwidth permits.

```
champion = wl.upload_model(champion_model_name, champion_model_file)
model2 = wl.upload_model(shadow_model_01_name, shadow_model_01_file)
model3 = wl.upload_model(shadow_model_02_name, shadow_model_02_file)
pipeline.add_shadow_deploy(champion, [model2, model3])
pipeline.deploy()
pipeline.logs()
pipeline.replace_with_model_step(0, model2)
```

# 3. Wallaroo Platform Components

The Wallaroo platform consists of three main components:

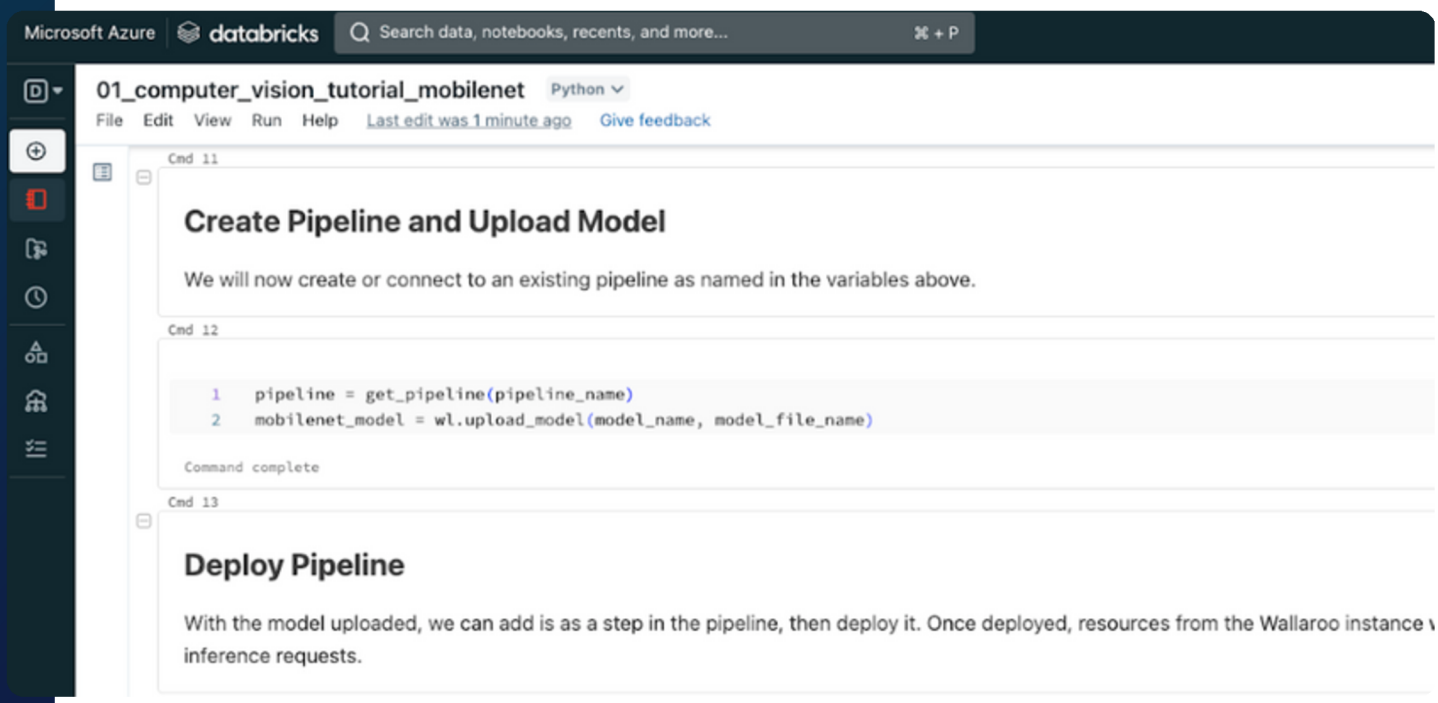
- Wallaroo Integration Toolkit
- Wallaroo Inference Engine
- Wallaroo Ops Center - Centralized Management & Observability



# 3.1. Wallaroo Integration Toolkit

Wallaroo's integration toolkit equips AI teams with everything they need to transition from prototype to production seamlessly and intuitively. Integrating with your existing AI toolchain it offers Notebook Integration with platforms such as OCI, Databricks, Jupyter, AWS, and Azure ML, and includes built-in data connectors for all major cloud providers.

The toolkit also integrates with both standard and custom orchestration stacks, such as Kubernetes. With ML and LLM Ops APIs, Wallaroo supports continuous integration, deployment, and optimization. Additionally, it provides Inference Logs API and SDK for business reporting integration, Inference Serving APIs for business applications, and custom model registry integration, including native MLFlow Model and Container Registry integration.



# 3.2. Wallaroo Inference Engine

The Wallaroo Inference Engine is optimized for high-performance in both model and data processing, supporting both batch and real-time workflows. It offers flexible deployment, scaling effortlessly from Raspberry Pi to large Kubernetes clusters, and easily transitions between x86, Arm64, Power10, and GPU architectures with simple configuration.

Achieving fast, efficient inference performances requires optimizing the serving and data-processing aspects of the inference server as well as the actual inference computations. Wallaroo Inference Server does both.

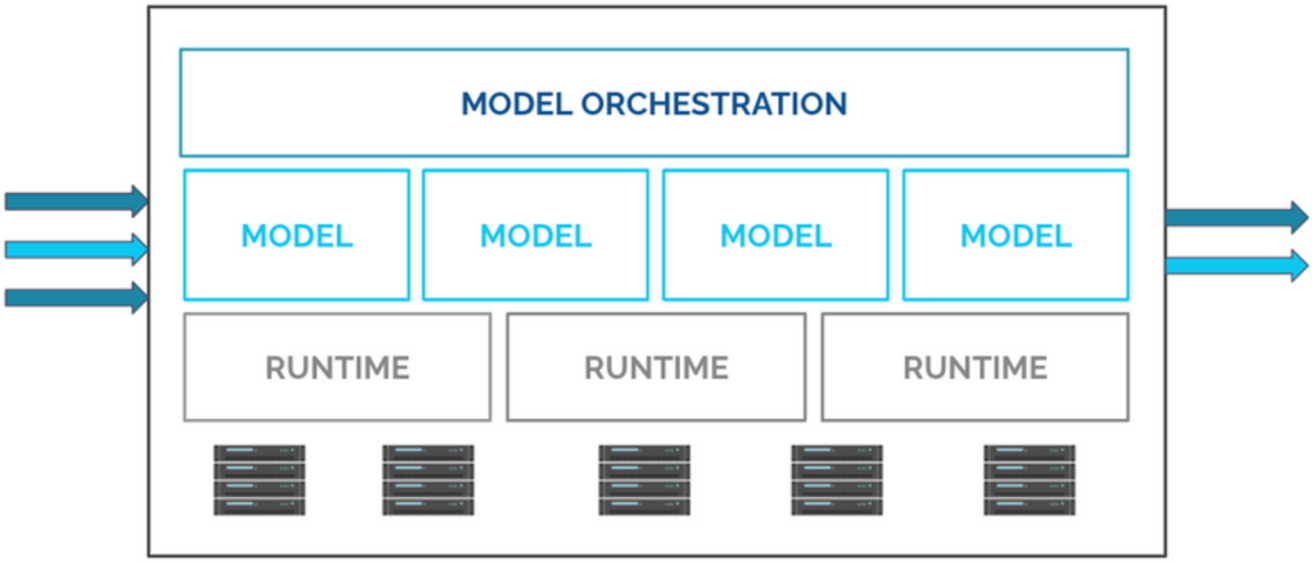
The Wallaroo Inference Server provides an optimized, multi-threaded network server and orchestration layer that is responsible for handling the in-bound requests, unpacking and preparing the incoming data for the model runtimes, and then invoking the models.

The inference engine is written in the Rust Language, a modern, high-performance language that enables detailed control over aspects such as memory management to reduce data copies and provides strong concurrency support to take advantage of modern, multi-threaded processors. Compared to typical Python implementations, this reduces overhead by 10-100x in some benchmarks.

Optimizing the actual inference computations means leveraging the best open-source and proprietary runtimes available. Simply using an open-source runtime, however, may not yield optimal results as many hardware manufacturers provide customized builds optimized for their hardware.

Wallaroo partners with industry-leading hardware vendors to integrate the best runtimes available and verify performance on these hardware platforms to allow AI teams to deploy with confidence.

Finally, in many applications the cost of CPU time will pale in comparison to the time and cost of the human effort to deploy the model. The Wallaroo Inference Engine provides an inference hardware abstraction layer that allows teams to focus on building models instead of dealing with hardware-specific details.



# Performance Benchmarks

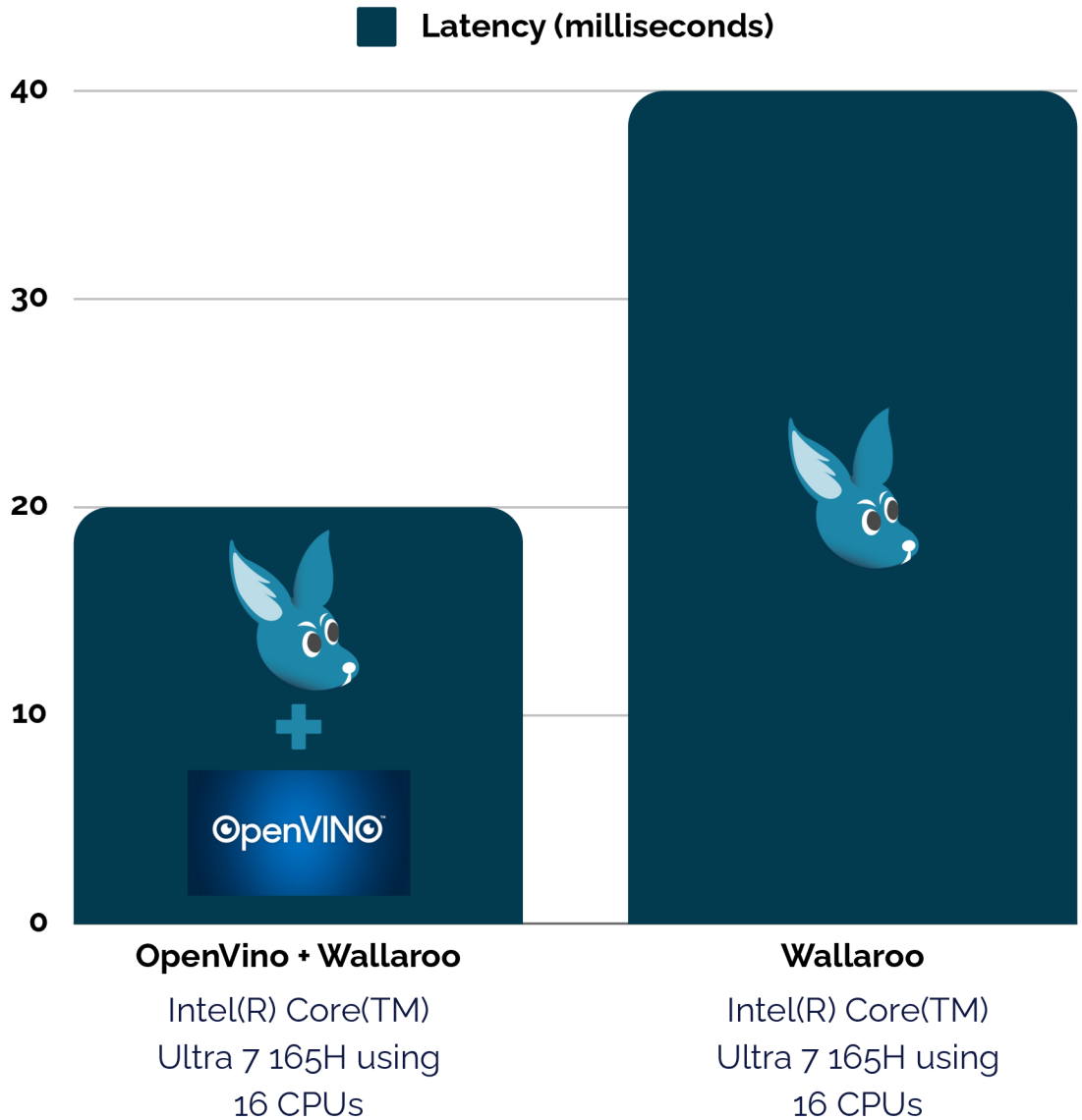
## Wallaroo + Intel OpenVINO

Wallaroo + Intel OpenVINO Benchmark Results

- ~2X faster inference time for optimized Wallaroo.AI + OpenVINO with ONNX vs. Wallaroo ONNX deployment without OpenVINO

Test scenario

- Computer Vision (CV) Resnet50 Model tested on Intel dev cloud



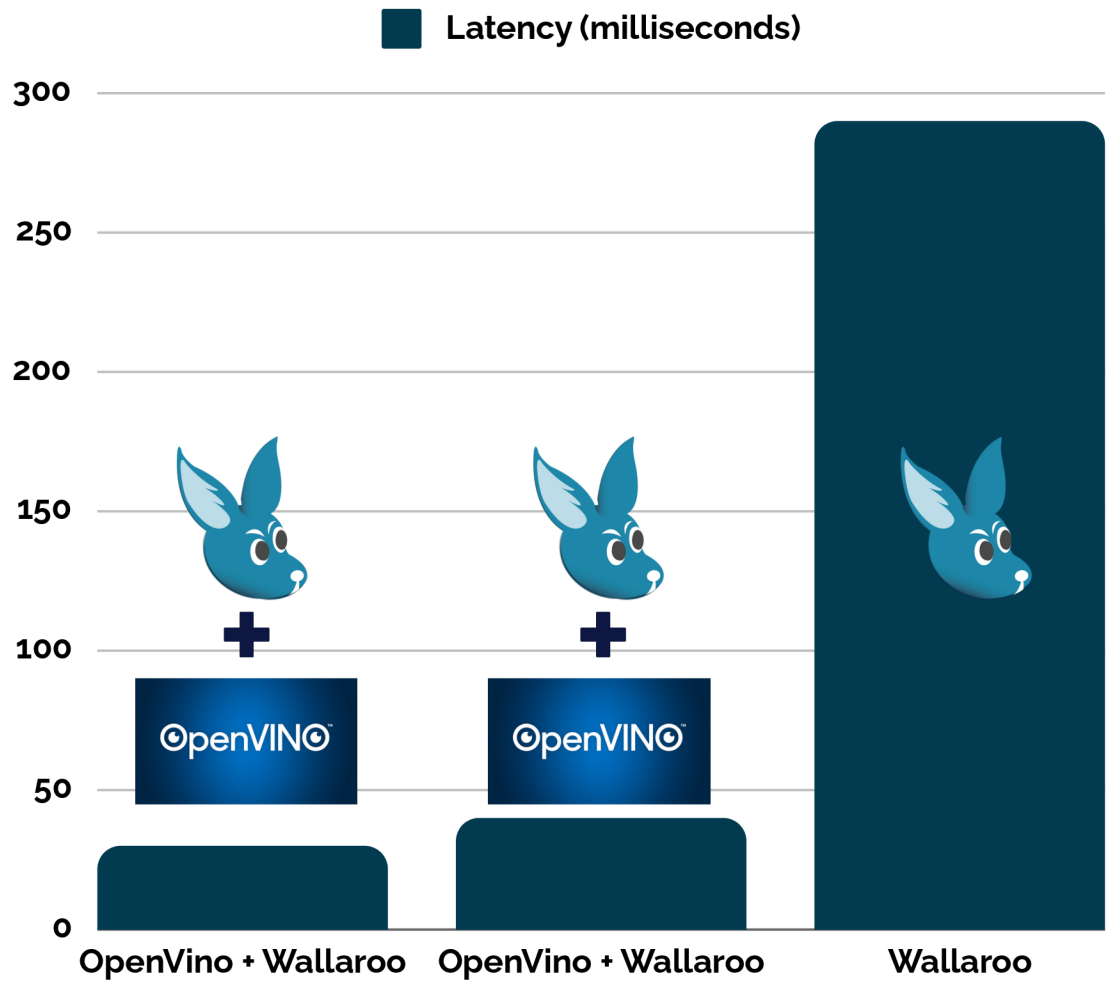
## Wallaroo + Intel OpenVINO

Wallaroo + Intel OpenVINO Benchmark Results

- ~10X faster inference time for optimized Wallaroo.AI + OpenVINO on intel GPUs

Test scenario

- Computer Vision (CV) YOLOv8n Model tested on Intel dev cloud




| Intel(R) Graphics [0x7d55] (iGPU) | Intel(R) Data Center GPU Flex 170 (dGPU) | Industry Standard GPU 16GB RAM |
|---|---|---|

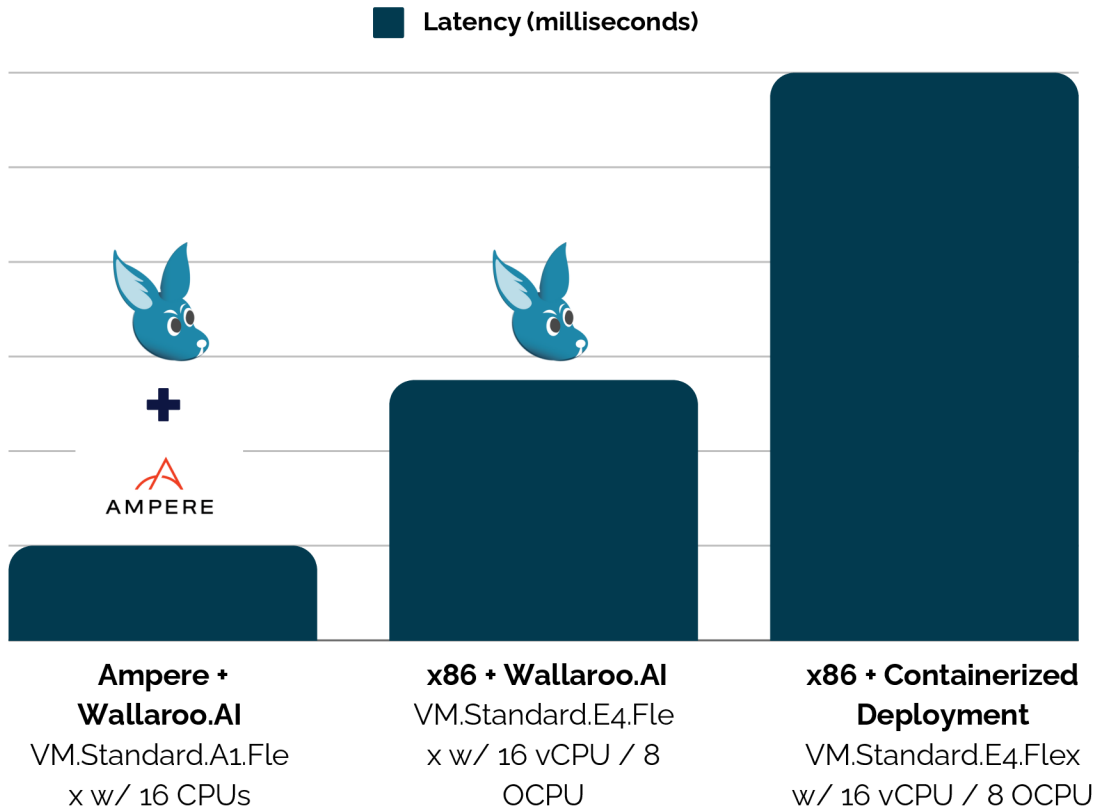
## Wallaroo + Ampere Benchmark Results

- Fastest inference time was optimized solution of Azure ARM + Wallaroo.AI (result A)
- x86 + Wallaroo.AI (result B) was significant improvement vs. x86 deployment (C)
- ~6X faster inference time for optimized Azure ARM + Wallaroo.AI + custom Arm ONNX (result A) vs. x86 deployment (C)
- Greater efficiency, reduced energy consumption, and lower cost per inference

Testing Platform

- Computer Vision (CV) Resnet50 Model tested on Azure
- Similar Results on Oracle, GCP.

7X faster & lower-cost than AWS Graviton



# Wallaroo Ops Center

## Centralized Management & Observability

Wallaroo's Ops center ™ provides AI teams an intuitive collaborative space to simplify and automate model packaging for deployment while enabling comprehensive monitoring, management and governance of all models and inference pipelines.

The Wallaroo Model Ops Center leverages a native integration with Kubernetes to install and integrate in a wide variety of AI production environments; compute clusters hosted by any of the major Cloud Providers, on-prem data centers, or even single servers.

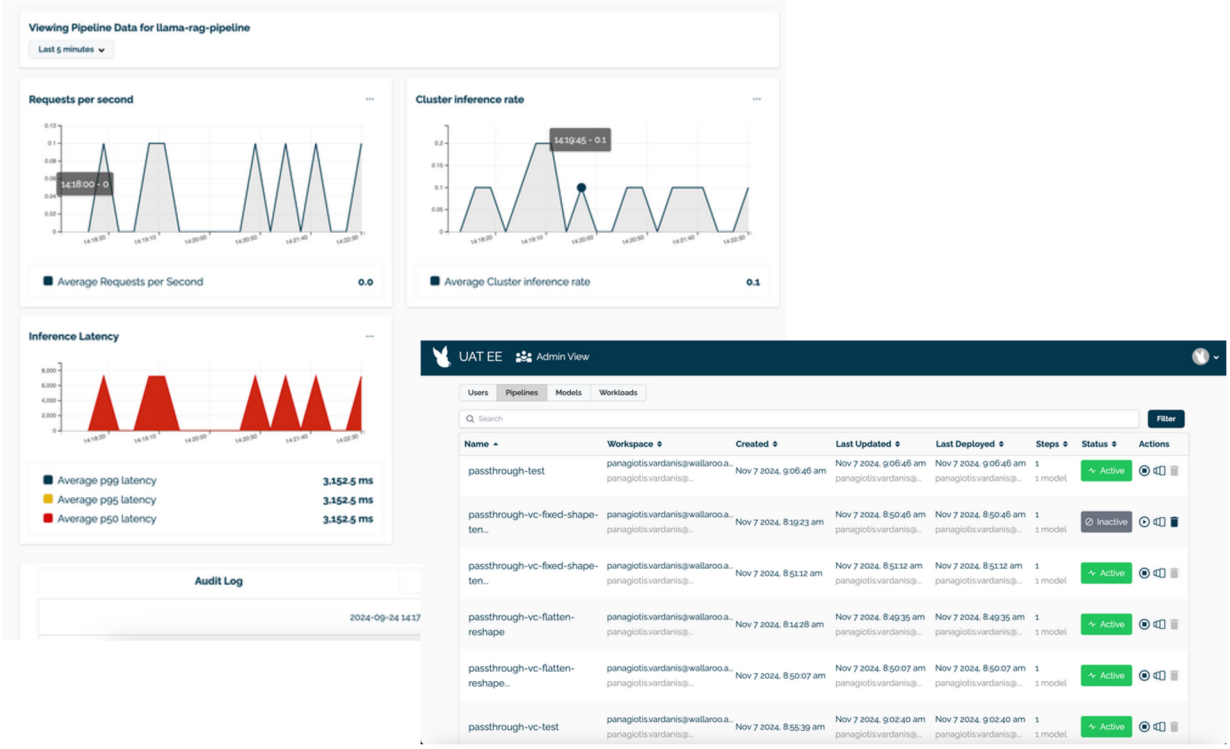
Kubernetes provides a robust, standardized environment for managing microservices deployed to heterogeneous hardware resources. Kubernetes enables three key aspects of Wallaroo:

- **Unified Platform:** Teams and organizations get a familiar, modern SaaS-like user experience with on-demand Python development environments, authentication that can be managed locally or integrated with existing enterprise identity providers, and a cohesive MLOps/LLMOps toolkit to package, deploy, observe and maintain inference pipelines in production environments.

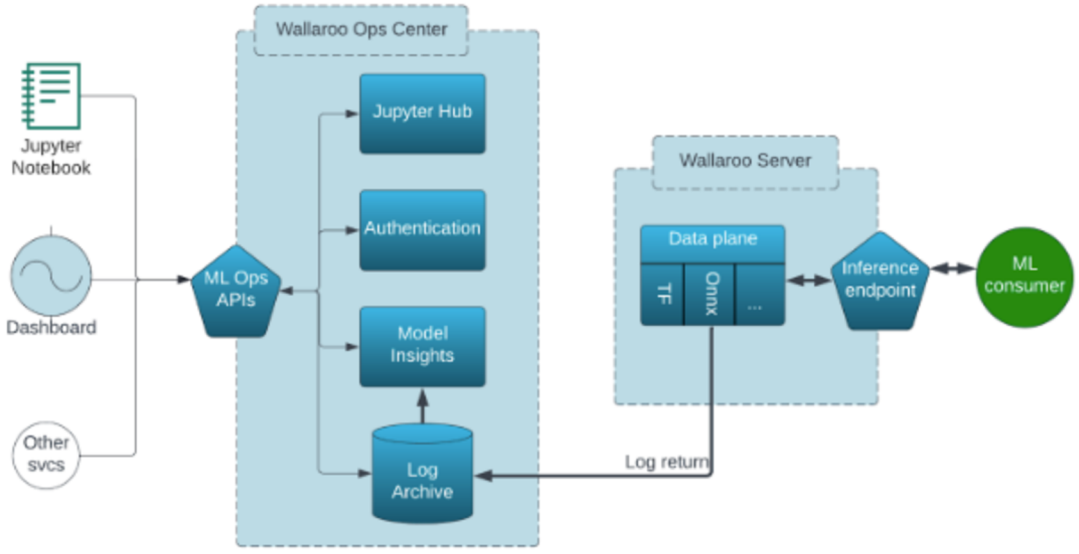
- **Automation & Reliability:** Inference pipeline deployments are “production-grade”, with features such as load-balancing, automated recovery, and load-based auto-scaling, and can be deployed to the range of hardware platforms available to the cluster to allow each application to run on the appropriate hardware.

- **Flexibility & Security:** Wallaroo installs into an organization's cloud tenancy or on-prem hardware to ensure sensitive data and models stay within the organization’s security boundary; no data or models are ever sent to external servers.


Wallaroo Ops Center - Centralized Model Management and Observability




The Wallaroo Ops Center comprises multiple services that, integrated together, provide a seamless enterprise-grade platform experience. Just a few of the services, described below, include authentication and user management, model registry services, an interactive Python environment, and AI/ML workload automation.

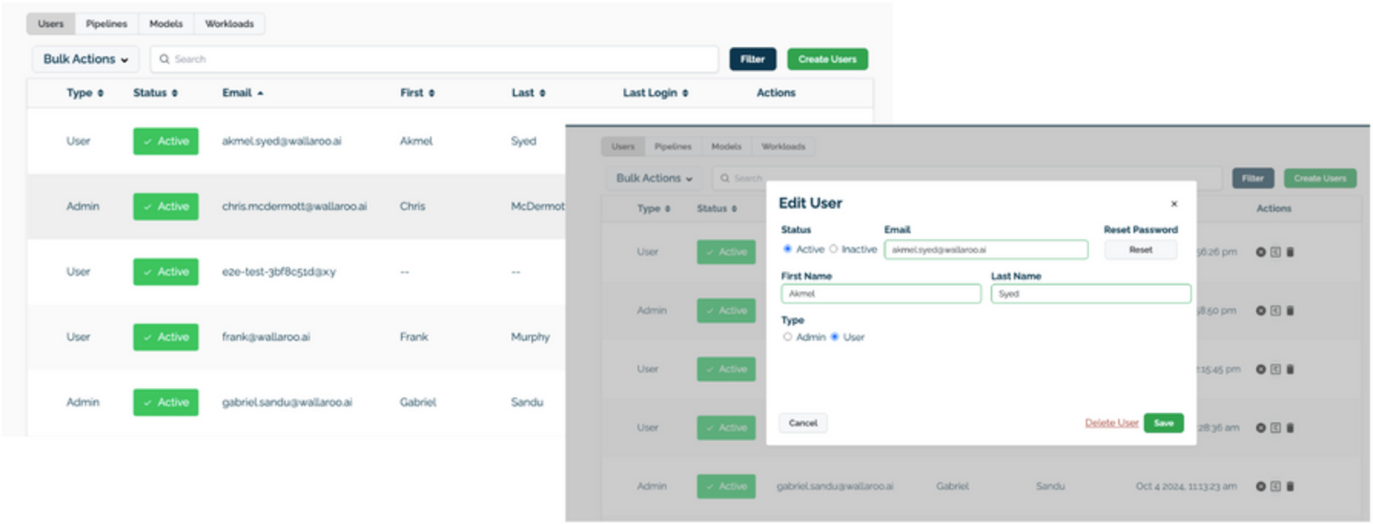



Wallaroo Ops Center - Architecture

## Authentication & User Management

Wallaroo builds on top of the well regarded open-source package “KeyCloak” to provide a flexible authentication system that users can manage themselves or can be integrated with an enterprise's existing identity provider.

The Wallaroo dashboard administrator page provides an easy-to-use interface for teams to self-manage access, team members, and roles. Larger organizations can avoid duplicate user management and instead delegate authentication to existing identity systems, allowing an enterprise’s IT department to centrally manage and federate access.

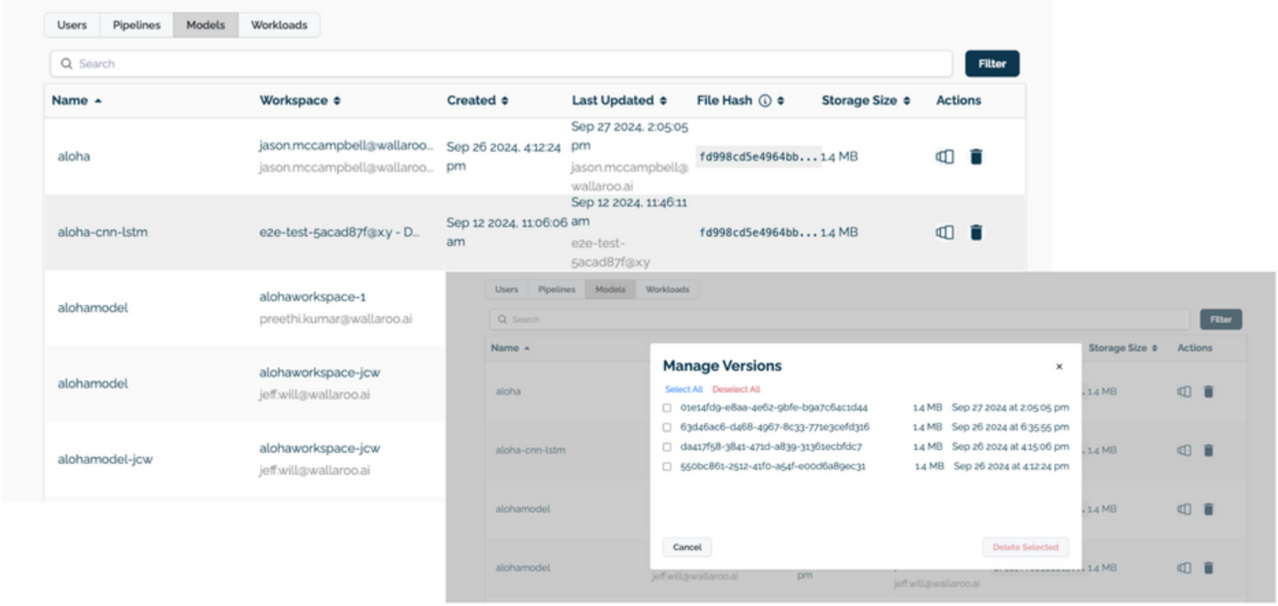



Wallaroo Ops Center - User management

## Model Registry Services

Wallaroo provides an integrated model registry for storing and managing the models, and model versions, in the system. Audit logs track and record changes to every model and inference pipeline.

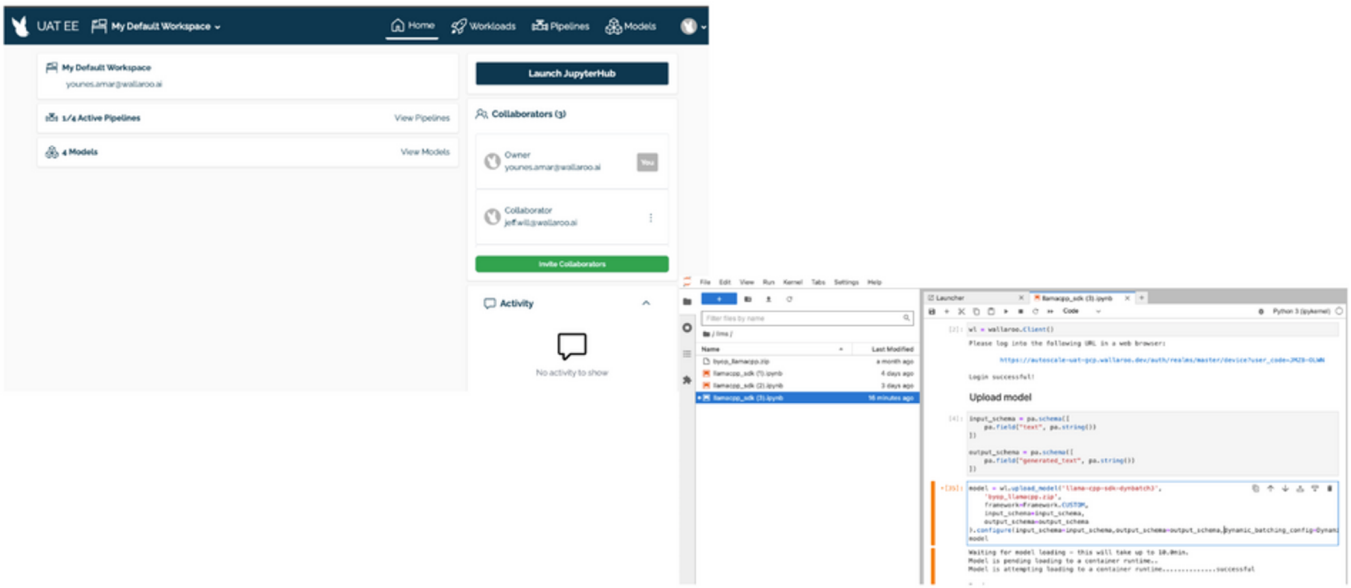
The Wallaroo Ops Center also integrates with external model registries. For example, Wallaroo provides native integration with the MLFlow registry, a popular open source tool.




Wallaroo Ops Center - User management

## Interactive Python Environment

The Wallaroo Ops Center provides an interactive Python notebook environment that is a staple of data science development. The notebooks offer easy access to the Wallaroo Python SDK (see Wallaroo Integration Toolkit) and flexible, interactive environment for standing up inference pipelines and analyzing results. The Python environment scales dynamically to support as many concurrent users as needed.



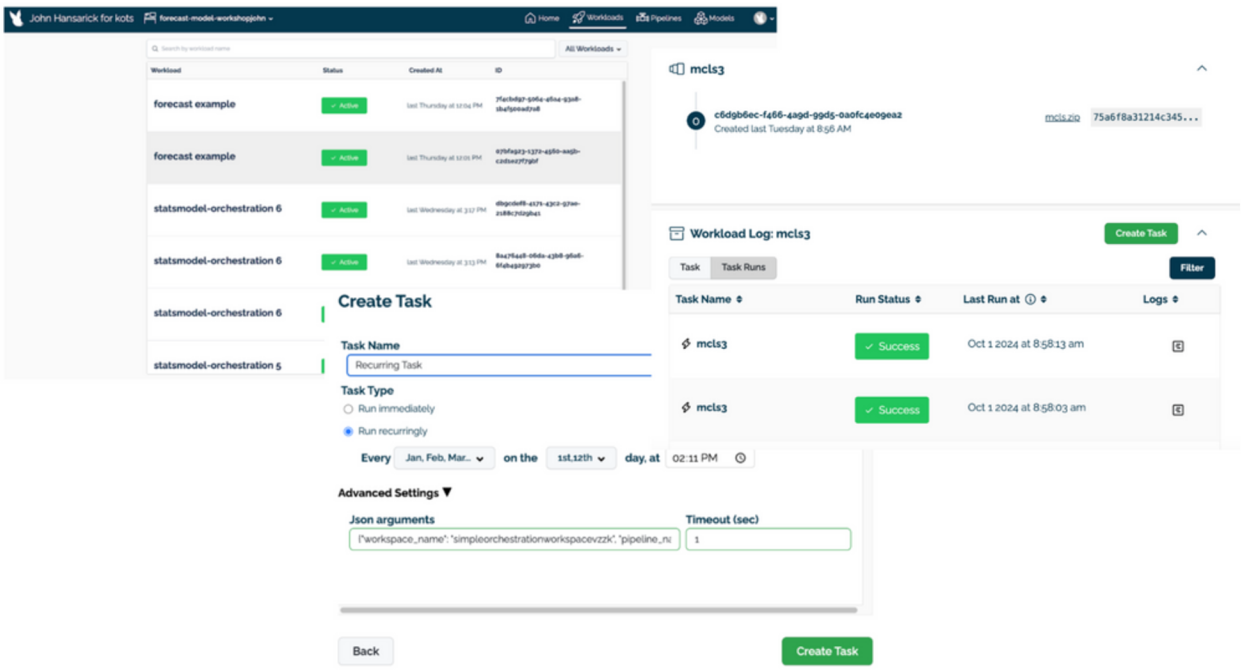

Wallaroo Ops Center - Interactive Python Environment

# AI/ML Workload Orchestration

The AI Workload Orchestration features allow teams to easily build complete data workflows on top of their inference pipelines. Workloads can query data sources, integrate with event systems (Kafka, RabbitMQ, etc), and store results to tools such as business analytics environments.

Workloads can be run on demand, on a schedule, or run as a service that continually receives input until terminated. One common use case is generating forecasts. The workload must pull results from many SKUs across all production facilities and run them through the pipelines.

Workload orchestrations are a good example of the benefits of a tight integration with Kubernetes with an easy-to-use interface: teams can quickly prototype AI-enabled workflows and then quickly scale both the workload and pipeline resources to handle demanding production jobs.




Wallaroo Ops Center - Workload Orchestration

# 4.Why Wallaroo

The Wallaroo AI Inference Platform enables enterprise AI teams to quickly and efficiently operationalize AI at scale through an end to end ML and LLM lifecycle from deployment, and ongoing model management with full governance and observability while realizing optimal performance scaling across Ampere, x64 & GPU architectures in the cloud, on-prem and at the edge.

- **Easily scale number of deployed models by 10X with a lean AI team**
- **Free up 40% of AI team's time, use up to 80% less inference infrastructure**
- **Continuously observe, optimize AI in production to get to value 3X faster**

By eliminating delays, reducing costs, and enhancing operational efficiency, the Wallaroo AI inference platform allows AI teams to focus on high-value tasks with maximum business impact.

# 4.1 Customer Testimonials

REALPAGE®

*"Wallaroo provides a lot more flexibility for deploying models than our current hot-swap method with AWS and GCP Vertex. It allows a much greater range of models. The speed to implementation of features that we didn't even know we needed has been great! It's been a great experience using Wallaroo."*

**- VP of Data Science**

AFRL

*"Wallaroo has demonstrated their AI/ML Platform to our researchers who believe this platform—with its interoperable and integrated architecture combined with our capabilities in partnership—can be deployed in the cloud and at the edge to deliver a game changing information advantage for critical space and military decision-making."*

***- Enrico Pontelli - Dean of the New Mexico State University***

*Air Force Research Laboratory Space Vehicles Directorate (AFRL/RV)*


*"The mission of the United States Space Force (USSF) is to organize, train, and equip Guardians to conduct global space operations that enhance the way our joint and coalition forces fight, while also offering decision makers military options to achieve national objectives. To do this effectively, we must invest in AI and ML capabilities that can be deployed in the cloud and at the edge. Wallaroo has demonstrated their AI/ML Enterprise Platform, and I believe this platform - with its uniquely modern, interoperable, and integrated architecture - is positioned exceptionally well to deliver game-changing capabilities to the USSF."*

***- Dr. Joel Mozer - Director Science, Technology & Research***

*SPACEWERX*


*"Wallaroo has been extremely good and useful for our intended use-cases. It has reduced our model deployment cycle time. The orchestration is also effective in reducing human involvement in the production environment."*

***- Principle ML Engineer***

# 4.2. Example Use Cases

## Computer Vision & Edge AI

Edge computing enables faster processing, reduced latency, and greater control over data privacy and security by bringing computing processes closer to the source of data generation. As a result, the adoption of edge computing is expected to increase by significantly through the end of the decade across many industries.

Wallaroo delivers flexible and scalable AI operations to decentralized edge and multi-cloud environments for connected and air-gapped deployment scenarios, while managing and observing models centrally.

- **Speed and Ease:** Seamless auto packaging and publishing of AI pipelines for diverse edge and multi-cloud deployment targets.
- **Scalability:** Deploy, and orchestrate models securely to decentralized Edge and Multi-Cloud locations with operational continuity while maintaining centralized, repeatable model management and observability.
- **Efficiency:** Optimized for flexible model deployment to connected or air-gapped hardware and architecture environments with centralized management and observability without forfeiting data driven decision velocity.

**Video Demo: Wallaroo Edge AI inference lifecycle (deployment, monitoring and updates)**

## GenAI and Large Language Models (LLMs)

With the emergence of GenAI and services associated with it such as ChatGPT, enterprises are motivated to jump on the GenAI train and make sure they are not left behind in the AI adoption stampede.

However, AI adoption has been a bumpy ride for a great deal of organizations due to underestimating the time, effort and cost it typically takes to get effective, reliable, and robust LLMs into production.

When LLMs are deployed to production the output generated is based on the training data of the model at the time. The model will take the user input and generate a text response based on the information it was trained on. As time goes by the model will gradually go out of date which can result in inaccurate generated text, hallucinations, bias etc.

Wallaroo empowers enterprises to deploy, manage, and scale GenAI and LLM models running with ease and flexibility within their data and AI ecosystem, enabling faster time to value and optimal infrastructure utilization.

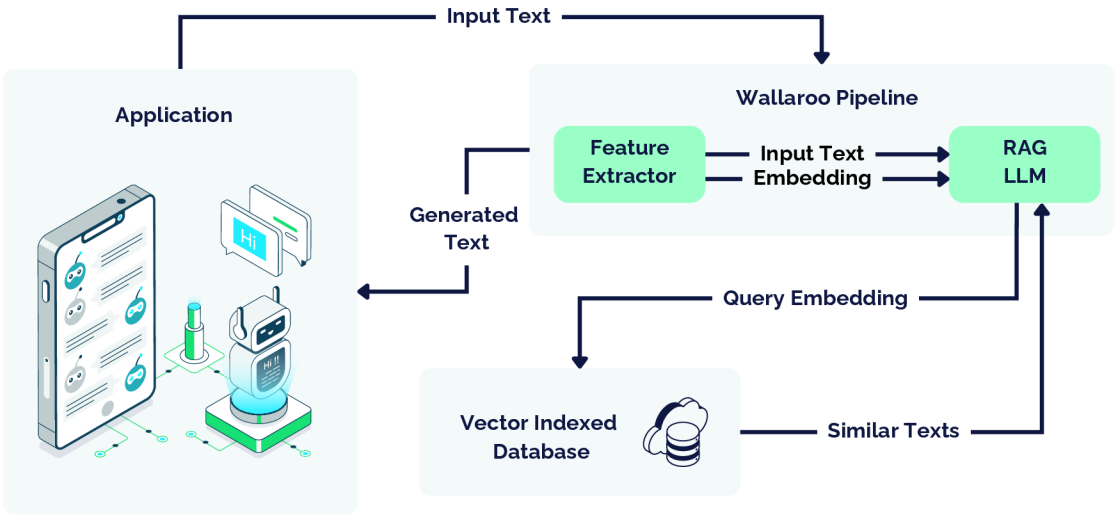
**So how do enterprises go about making LLMs accurate, relevant, and free of bias and hallucinations without having to constantly retrain the model?**

## LLM Automated Validation with RAG

Retrieval-Augmented Generation (RAG) are methods that helps LLMs to produce more accurate and relevant outputs, effectively overcoming some of the limitations inherent in their training data. RAG works by allowing the LLM to reference an authoritative knowledge base outside of its training data sources before generating a response.

The benefits of RAG:

- Enhanced reliability, accuracy, and performance of LLMs.
- Easily keep LLMs Up-to-date and free from hallucinations and toxicity.
- Avoid business risks and safeguard accurate and relevant outputs.




**Video Demo: Deploying LLM Inference Endpoints & Optimizing Output with RAG in Wallaroo**

## Automated LLM Monitoring

Wallaroo LLM Listeners ™ are used as native monitoring method for LLMOps to ensure that LLMs in production are robust and effective, by implementing monitoring metrics and alerts for potential issues such as toxicity, obscenity, harmful language, etc. to avoid risks safeguards accurate and relevant outputs.

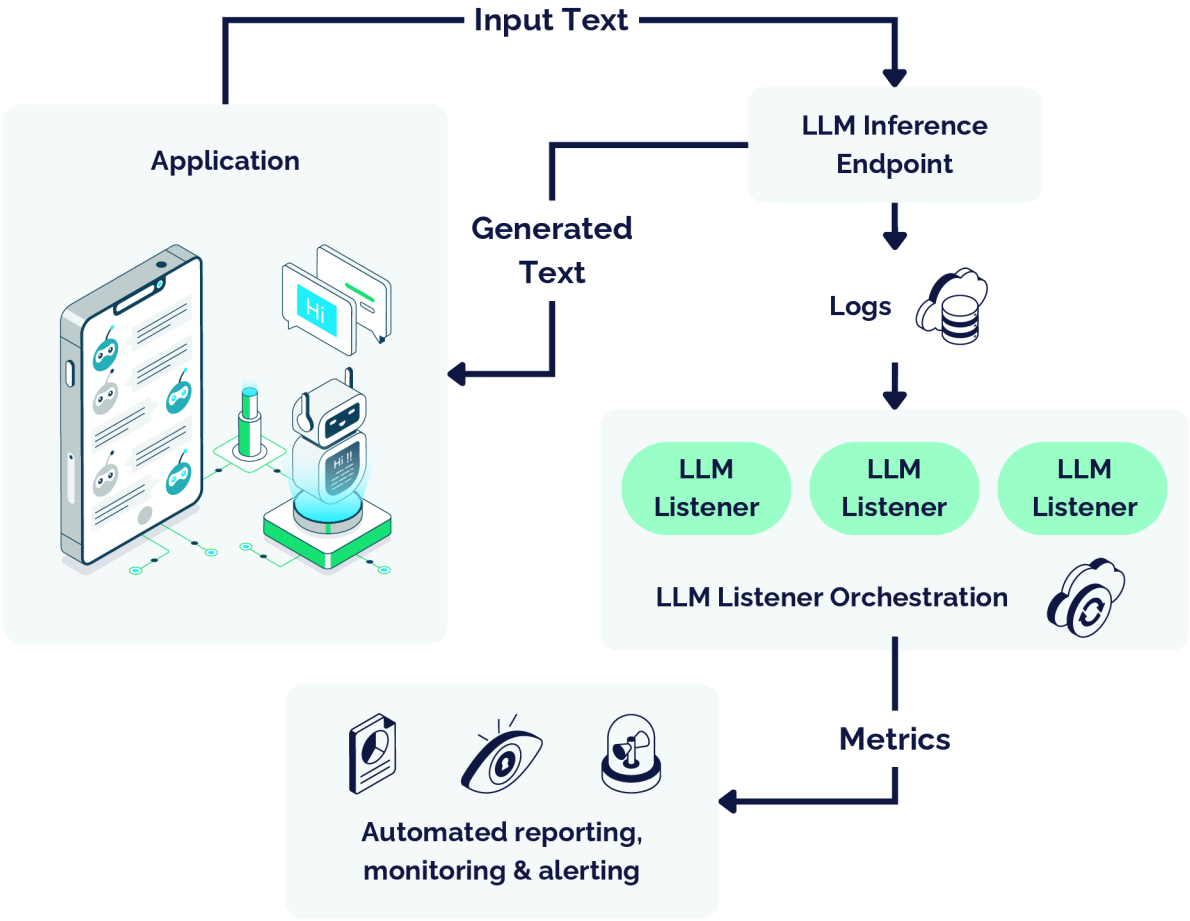
The Wallaroo LLM monitoring solution includes LLM deployed as part of a single inference pipeline in Wallaroo, as well as automated monitoring metrics and alerting with a vetted set of LLM listeners.

- Deploy the model of your choice on any infrastructure architecture to any endpoint with centralized automated management for scale and faster time to value.

- Inference logs from model deployments available in Wallaroo Dashboard.

- Automated orchestration to compute LLM scores on demand or on a schedule.

- Continuous monitoring and feedback loops with system alerts to reduce exposure to performance, bias, and privacy risks without disruption ensuring LLMs remain effective.




Wallaroo is actively working on building out a suite of these LLM Listeners and partnering with customers to build out listeners that are specific to their applications and use cases.

**Demo Video: Monitoring LLM Inference Endpoints with Wallaroo LLM Listeners**

# 5.Additional Resources

- Wallaroo.AI
- Wallaroo YouTube
- Wallaroo Product Documentation
- Blogs
- Contact