
AWS
re:Invent
DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

AIM313

High-scale performance optimization of serving multiple FMs

Giuseppe Zappia

Principal AI/ML Specialist Solutions Architect

AWS

Ram Vegiraju

ML Solutions Architect

AWS

Agenda

Generative AI model hosting challenges

Why fine-tune?

Benefits of LoRA and serving LoRA adapters at scale

What is SageMaker

Hosting multiple GenAI models

Performance optimizations through LMI container

Workshop agenda

Challenges of hosting ML/Foundation models

Performance

Host multiple FMs on the same endpoint. using multiple ML accelerators on the Instance.

MLOps

Establishing operational rigor in your ML strategy by monitoring, collaborating and actioning all initiatives in a centralized manner

Time To Market

Choosing the right model, deciding whether to train, fine-tune or consume, and the complexity of getting started

Go To Market

Monetizing your model by identifying new routes to market

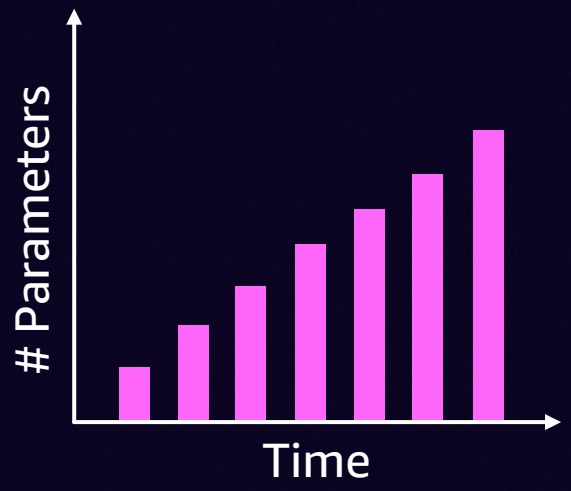
Cost Performance

Choosing the right model & compute options for efficient price performance

Hosting at Scale

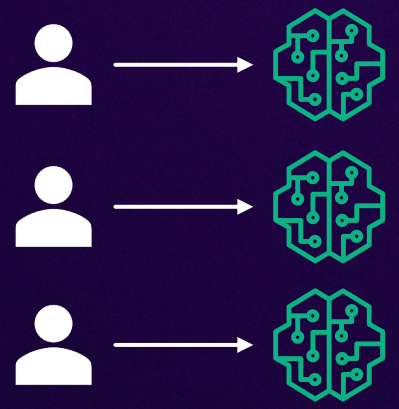
Managing your infrastructure and complexities such as endpoint management, autoscaling and deployment configurations

Emerging Trends in Deep Learning



Model sizes keep growing across domains; fine-tuning is the status quo

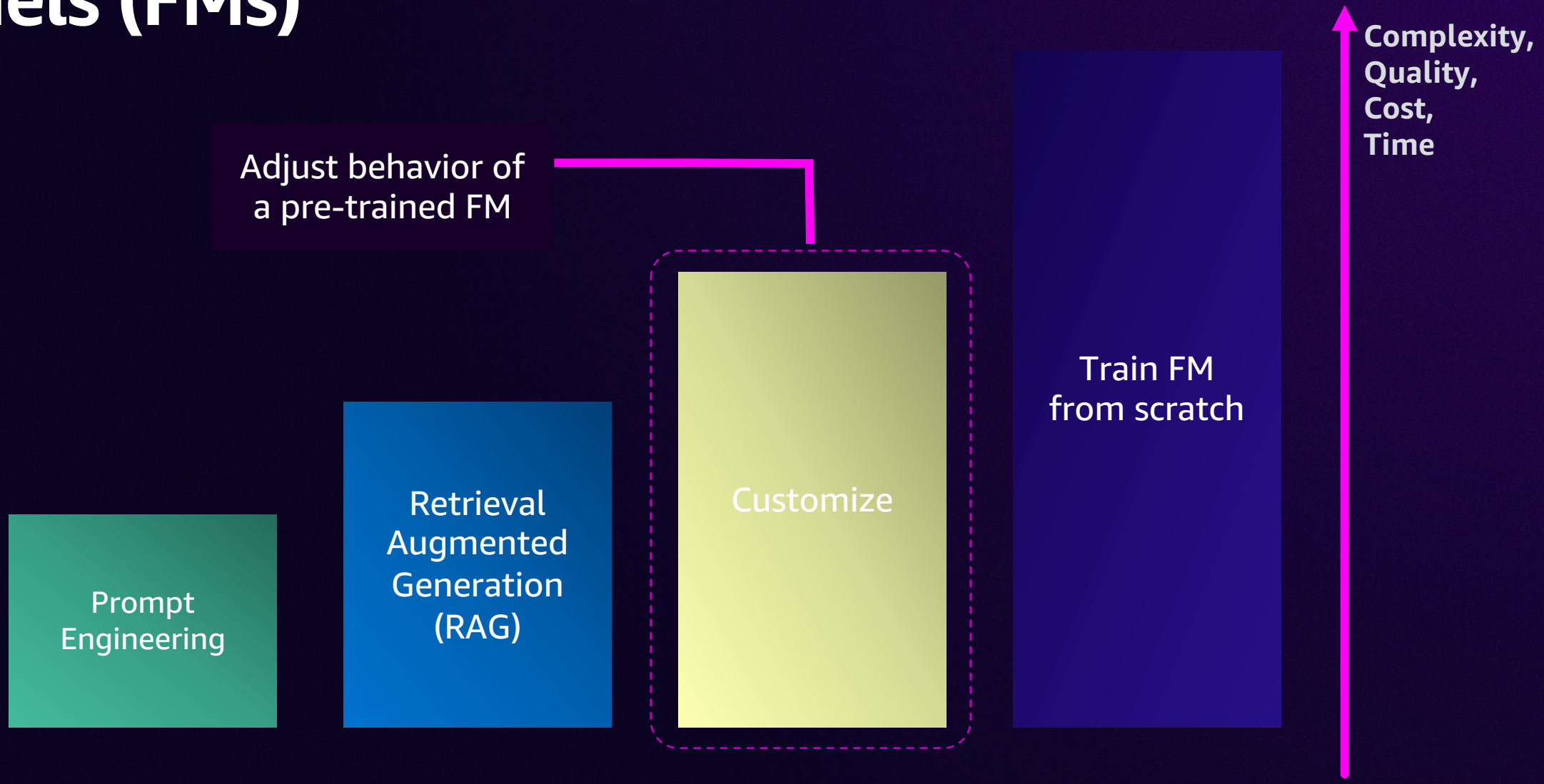
Multi-modal architectures are unlocking new applications



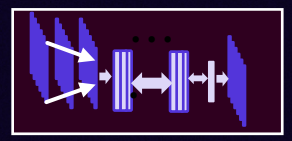
Proliferation of hyper-personalized models

aka “one model per customer” pattern

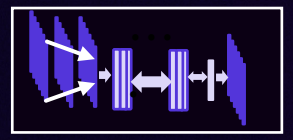
Common approaches for customizing foundation models (FMs)



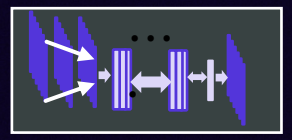
Scale up to 100’s



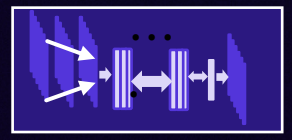
Code generation



Multi Lingual



Speech to Text



Summarization

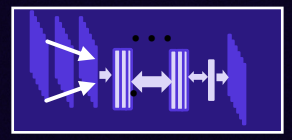
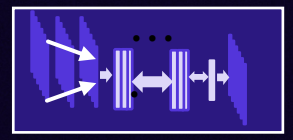
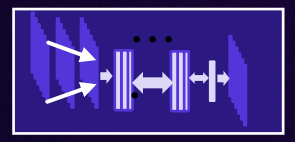


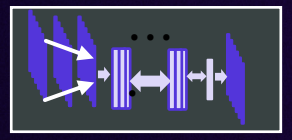
Image to text



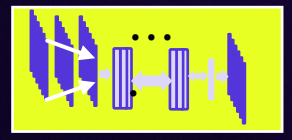
Text2Image



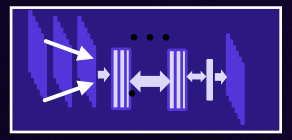
Model1



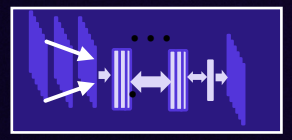
Model1



Model1



Model1



Model1

100’s of Base Models
100’s of Adapters
100’s of GPU’s

Cost grows linearly

PEFT & LoRA

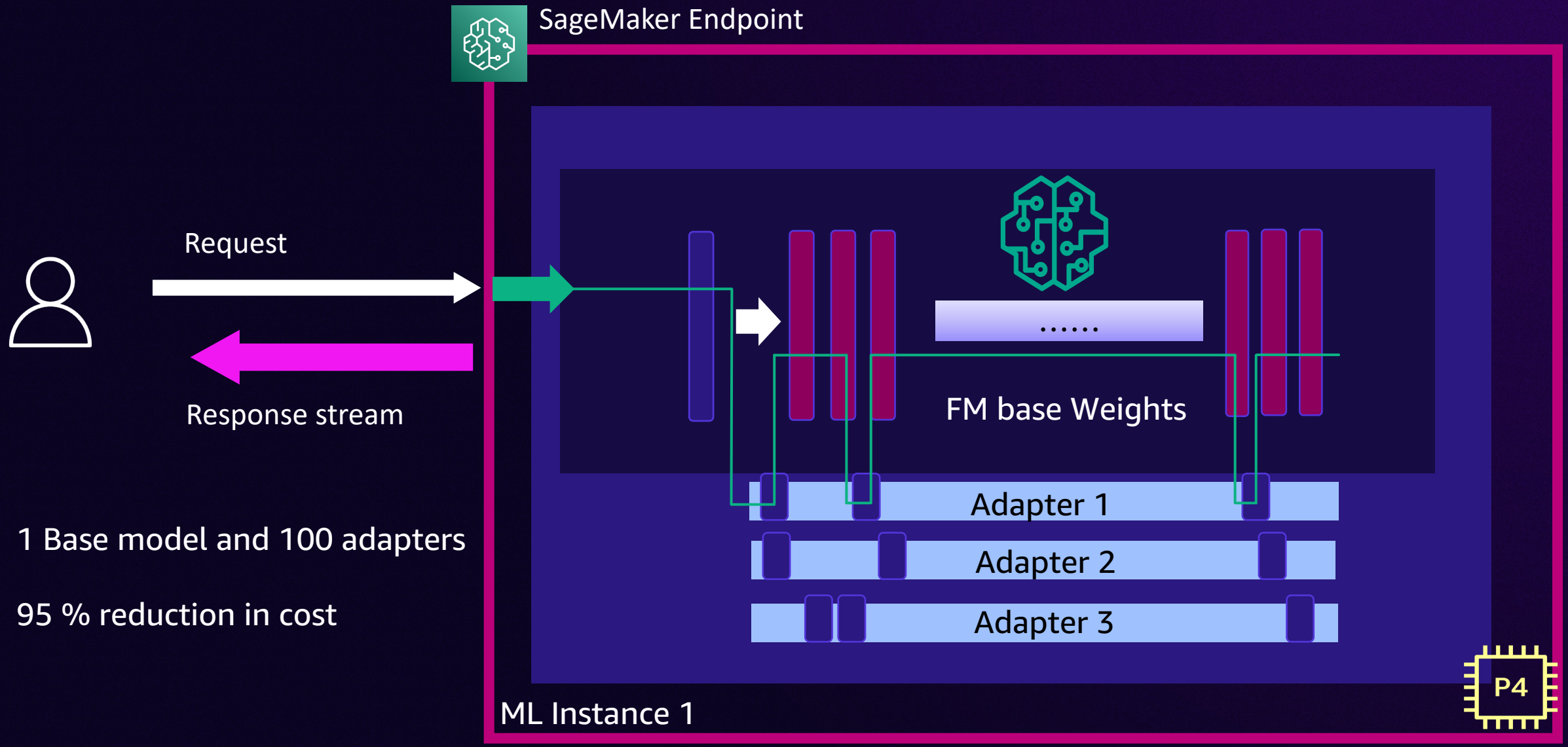
Why Use LoRA and PEFT Adapters?

- Reduce the computational cost of fine-tuning large language models
- Enable fast adaptation to new tasks or domains
- Preserve the model's general language understanding capabilities

When to Use LoRA and PEFT Adapters?

- Limited compute resources
- Fast adaptation
- Preservation of generalization
- Ability to retain base model

Let us Un-merge the Adapters



- 1 Base model and 100 adapters
- 95 % reduction in cost

Foundation Model Inference

Flexibility to deploy

your own Foundation Models (FMs)

100s of FMs at Scale

Customer-specific fine-tuned FMs

Multi-modal

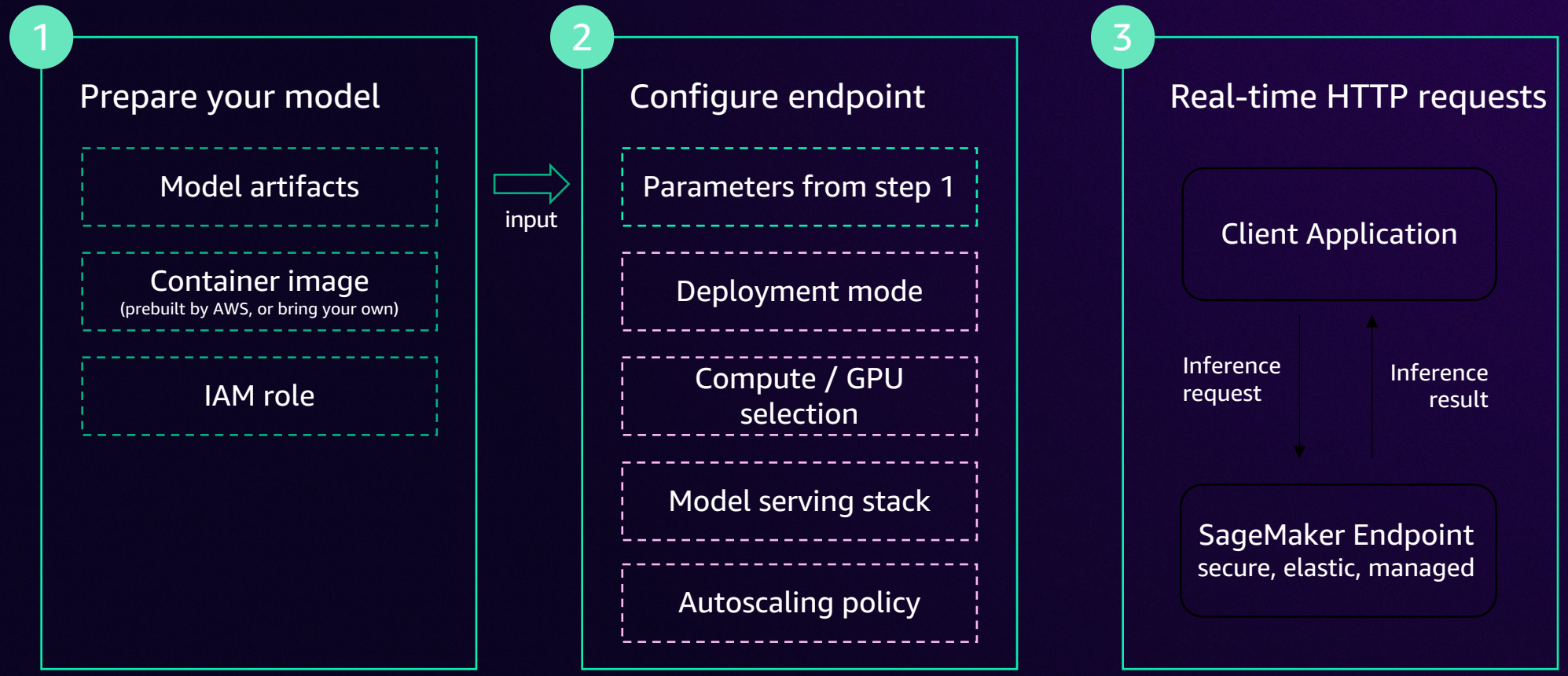
Video, Image, Audio, Code, Text, Documents

Price-performant Inference

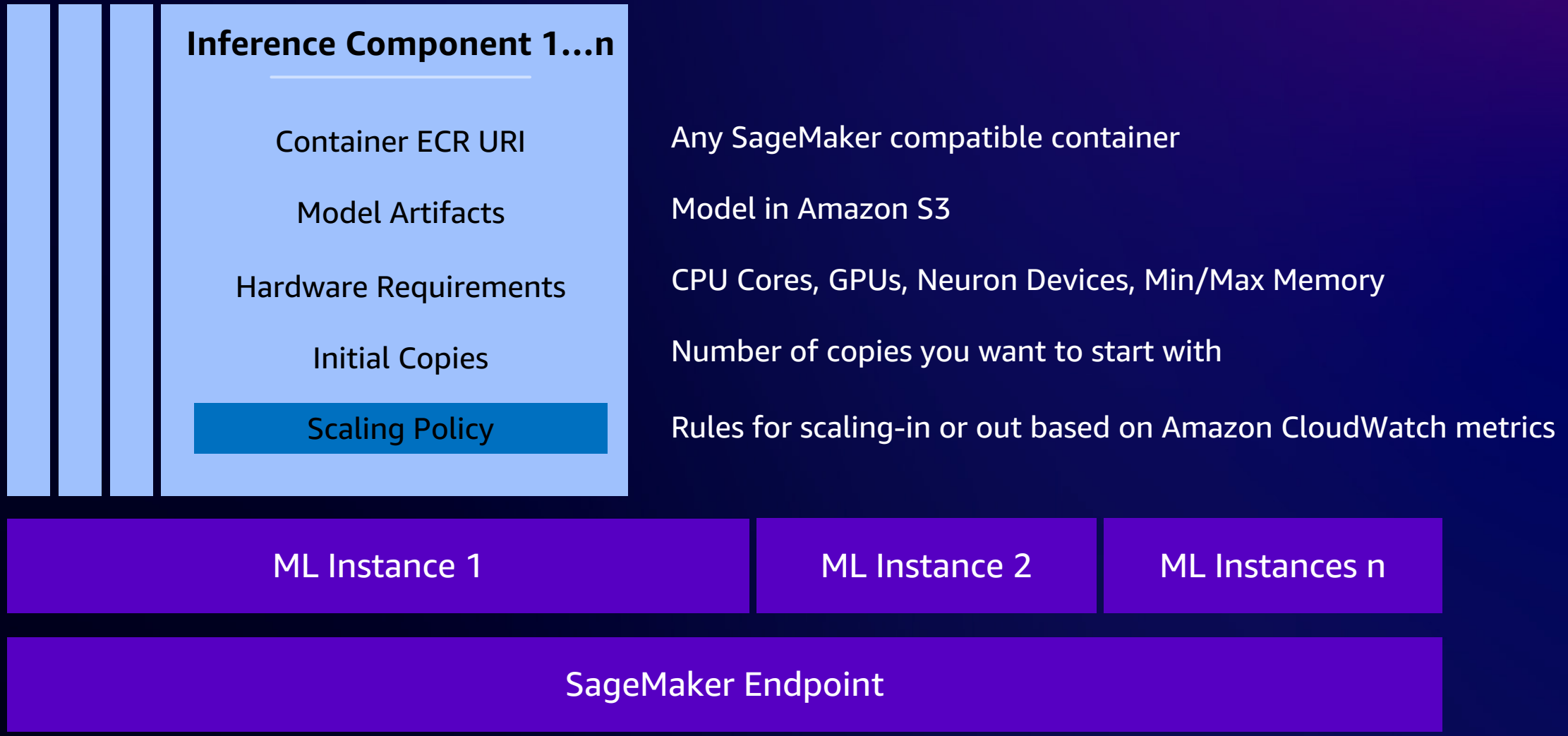
Any use case with best price-performance

How does Amazon SageMaker Inference work?

Example to illustrate “Real-time inference” configuration in 3 easy steps.



New model abstraction- Inference components



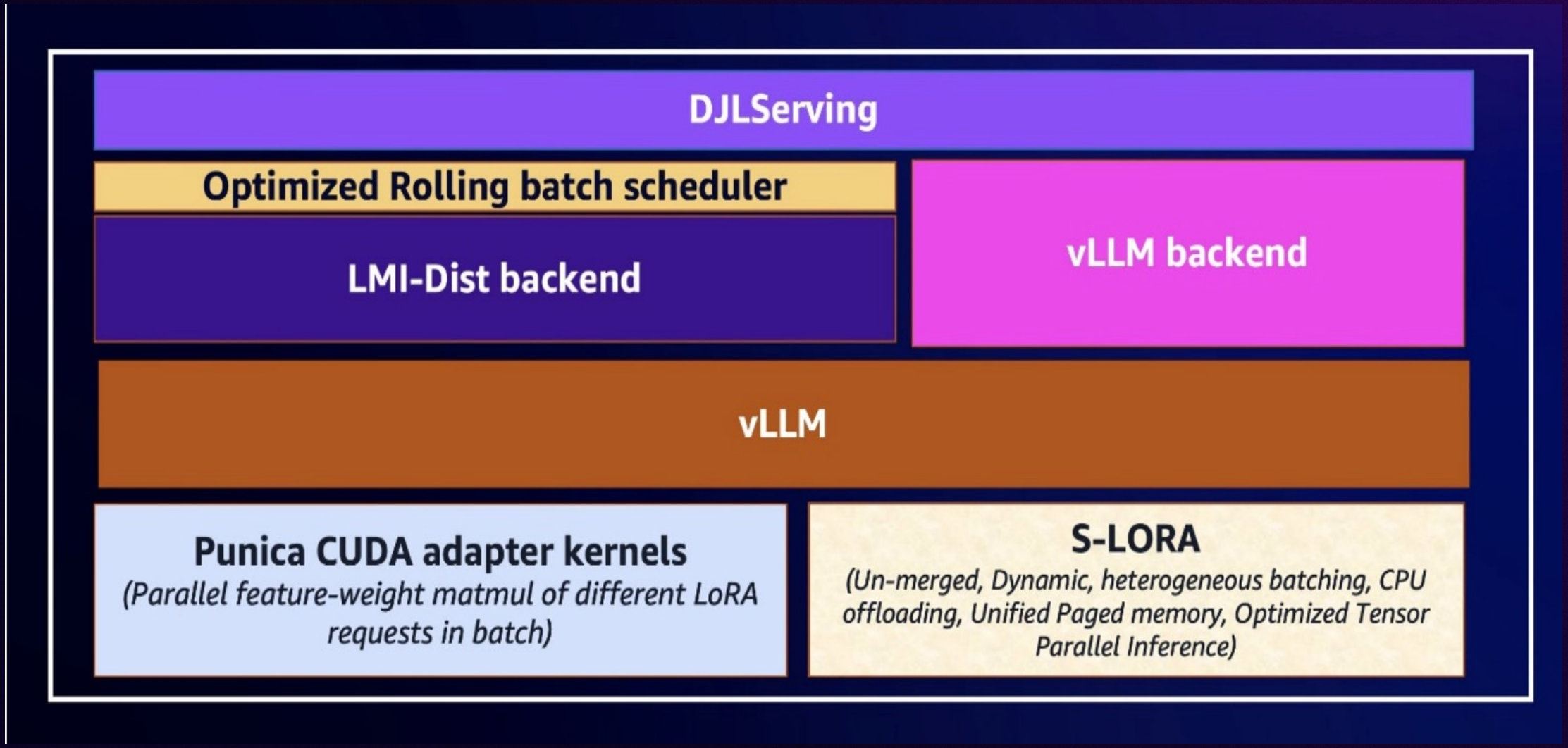
Large model inference (LMI) container on Amazon SageMaker

Large ML models

with 100 billion + parameters

- ✓ Faster model download time using s5cmd
- ✓ Supported by AWS and open source
- ✓ Low-code/no-code deployment
- ✓ Native integration with Inferentia/Trainium
- ✓ SageMaker multi-model, multi-container, batch, and serverless
- ✓ Pre-built optimized model parallel frameworks including vLLM, TensorRT-LLM, LMI-Dist, NeuronX-Transformers
- ✓ Pre-built foundation software stack including PyTorch, NCCL, and MPI

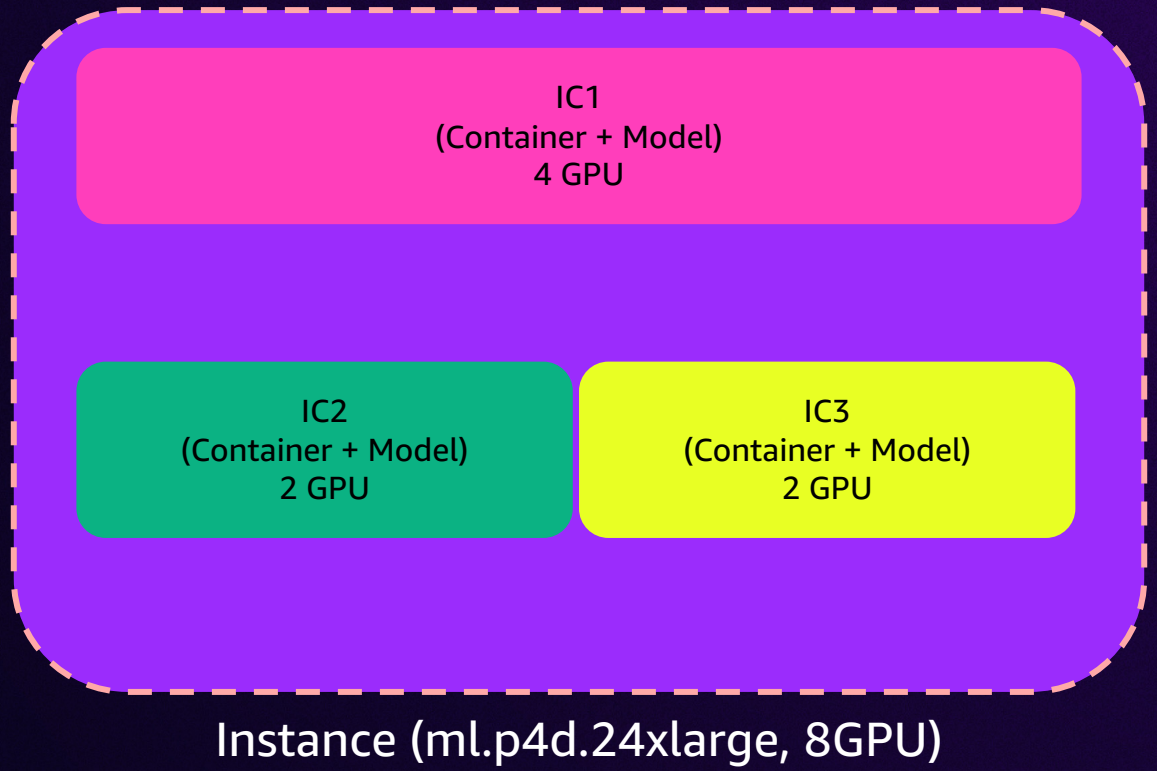
Model Serving Multi-Adapter LMI Container



Single/Fused Adapter Inference

Benefits:
- Fast, since everything is merged
- Still able to gain efficiencies during training

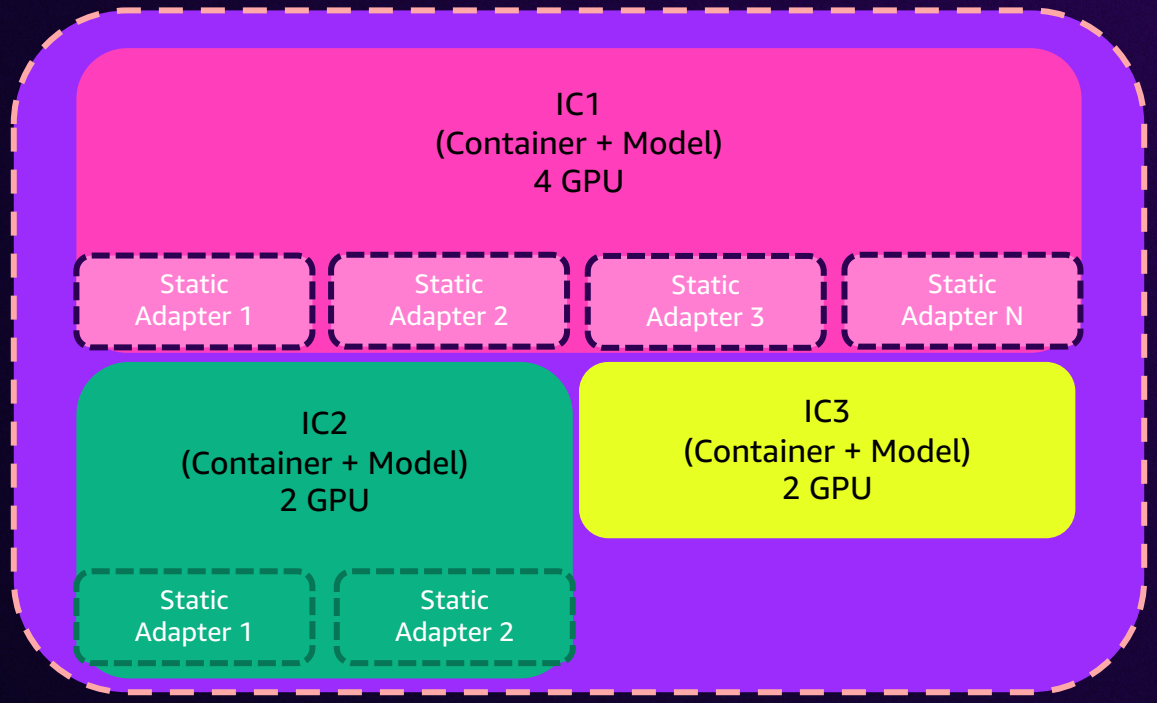
Tradeoffs:
- Cost



Multi Adapter Inference via Model Artifact

Benefits:
- Efficiency
- Cost-Optimization
- Can invoke multiple adapters in
- a single API call

Tradeoffs:
- Slightly slower than fused
- Rigid model package
- Difficult to manage individual adapters



Instance (ml.p4d.24xlarge, 8GPU)

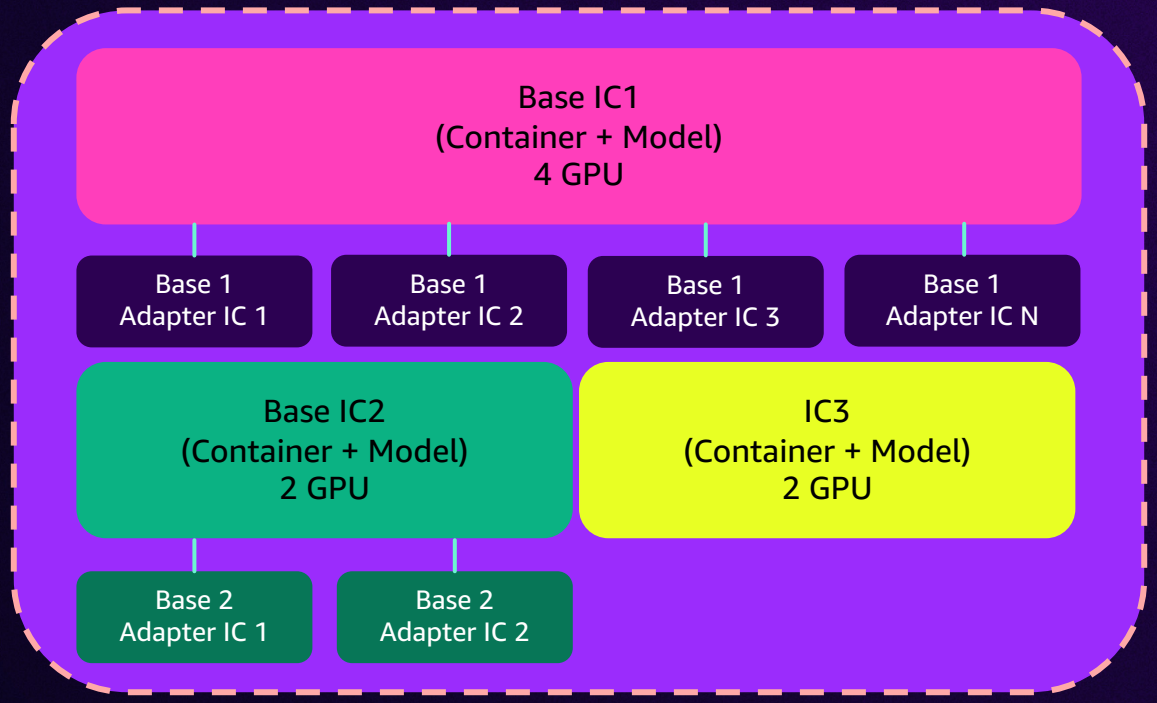
Multi Adapter Inference via Inference Components

Benefits:

- All of the previous benefits from static adapters
- Adapters can be independently managed

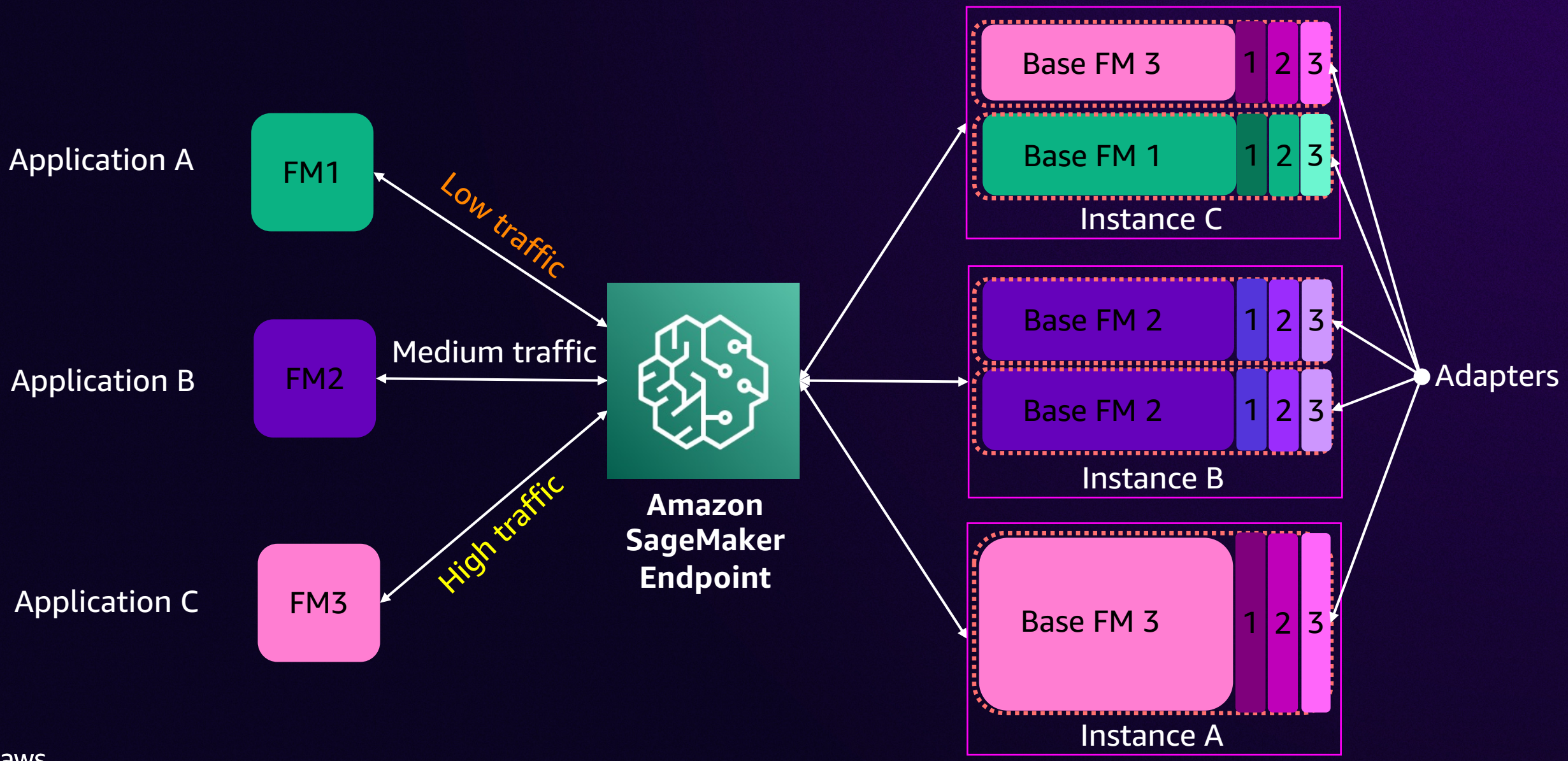
Tradeoffs:

- Slightly slower than fused
- API call only invokes a single adapter



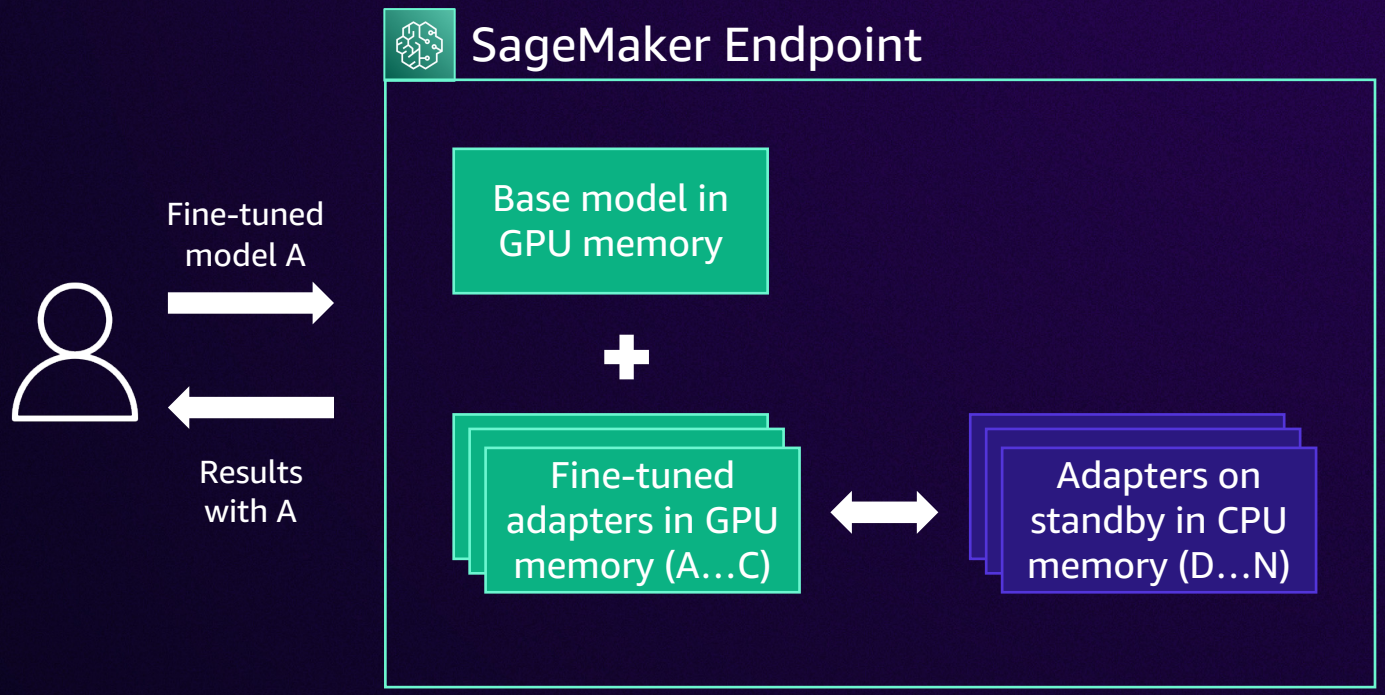
Instance (ml.p4d.24xlarge, 8GPU)

SageMaker Inference Components



Host hundreds of fine-tuned adapters

- Save costs
  - Host multiple base models with their own fine-tuned adapters on the same endpoint and instances
- Ease of use
  - Manage adapters on the endpoint with lifecycle APIs
  - Monitor usage of each fine-tuned adapter
  - Autoscale base model up and down in response to traffic
- Low overhead latency
  - <1ms overhead to use adapters in GPU memory
  - <10ms overhead to load adapters from CPU to GPU memory for inference



Workshop agenda

- Traditional hosting LoRA adapters with the Large Model Inference Container
- LoRA hosting using Inference Components and SageMaker efficient adapter hosting

Thank you!

Please complete the session survey in the mobile app