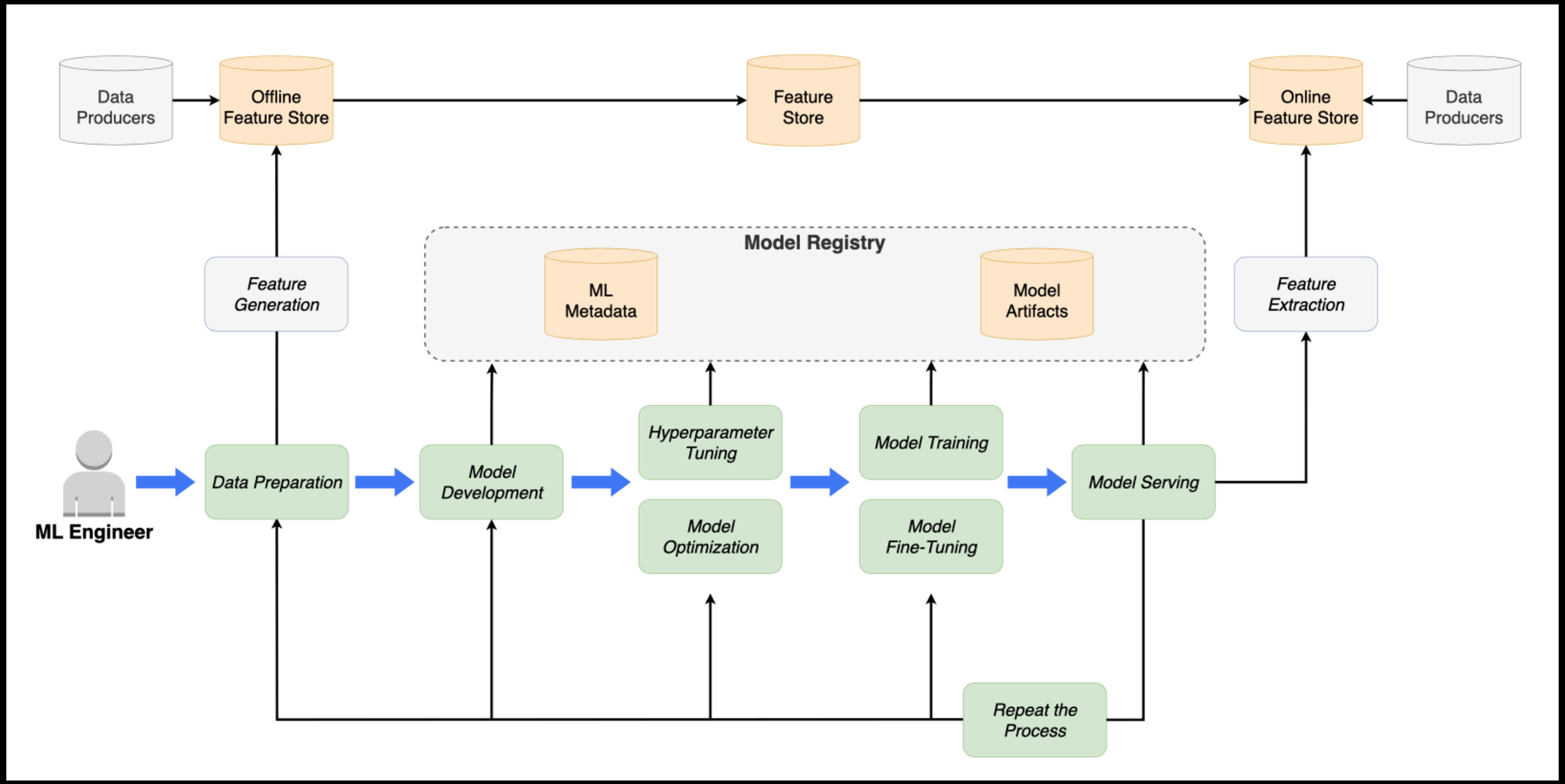
Cost Saving Strategies for Interactive AI Development

Cloud Native & Kubernetes AI Day ‘24

Shravan Achar, Engineer, Apple

AI / ML ecosystem is complex



Source: https://www.kubeflow.org/docs/started/architecture/#introducing-the-ml-lifecycle

How is AI Development different?

- AI development requires much larger compute resources
- Local environment often not sufficient
- Interactivity
- More collaborative environment

Compute Stack

Interactive Interface

Data Processing

Kubeflow

Machine Learning

Resource Scheduler

Container Platform

Optimizations

Application Layer

Dynamic Allocation

Distributed Training Strategies (FSDP)

Batching

Scheduling Layer

Fair Sharing

Gang Scheduling

Preemption

Resource Borrowing

Guaranteed Resources

Infrastructure Layer

Cloud Autoscaling

GPUs

Spot pricing

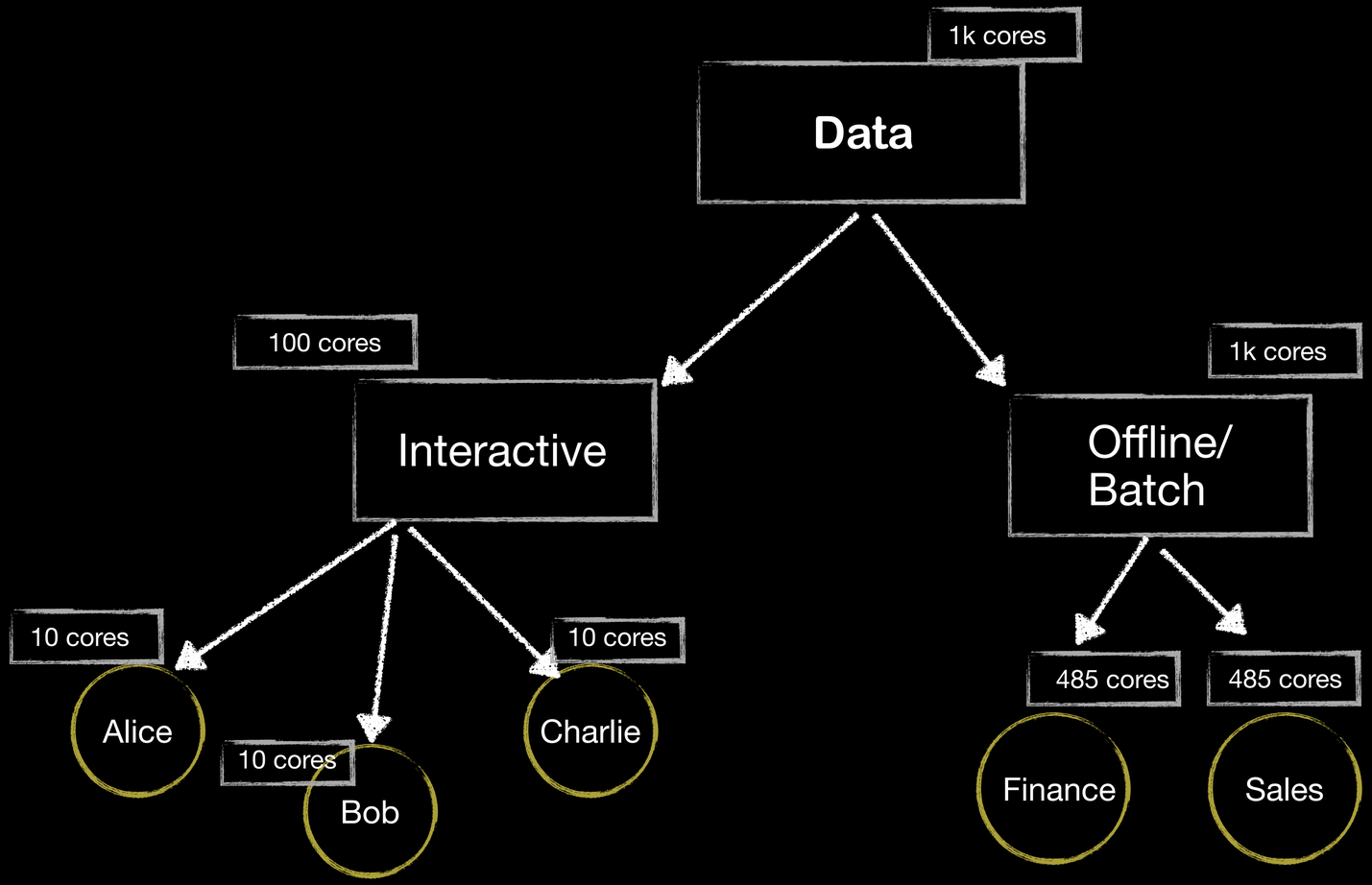
and more ...

Data Preparation

Imagine a team of data scientists working with data using Spark. They would benefit from

- Elasticity
- Fair sharing
- Autoscaling
- Guaranteed resources

Resource Structure

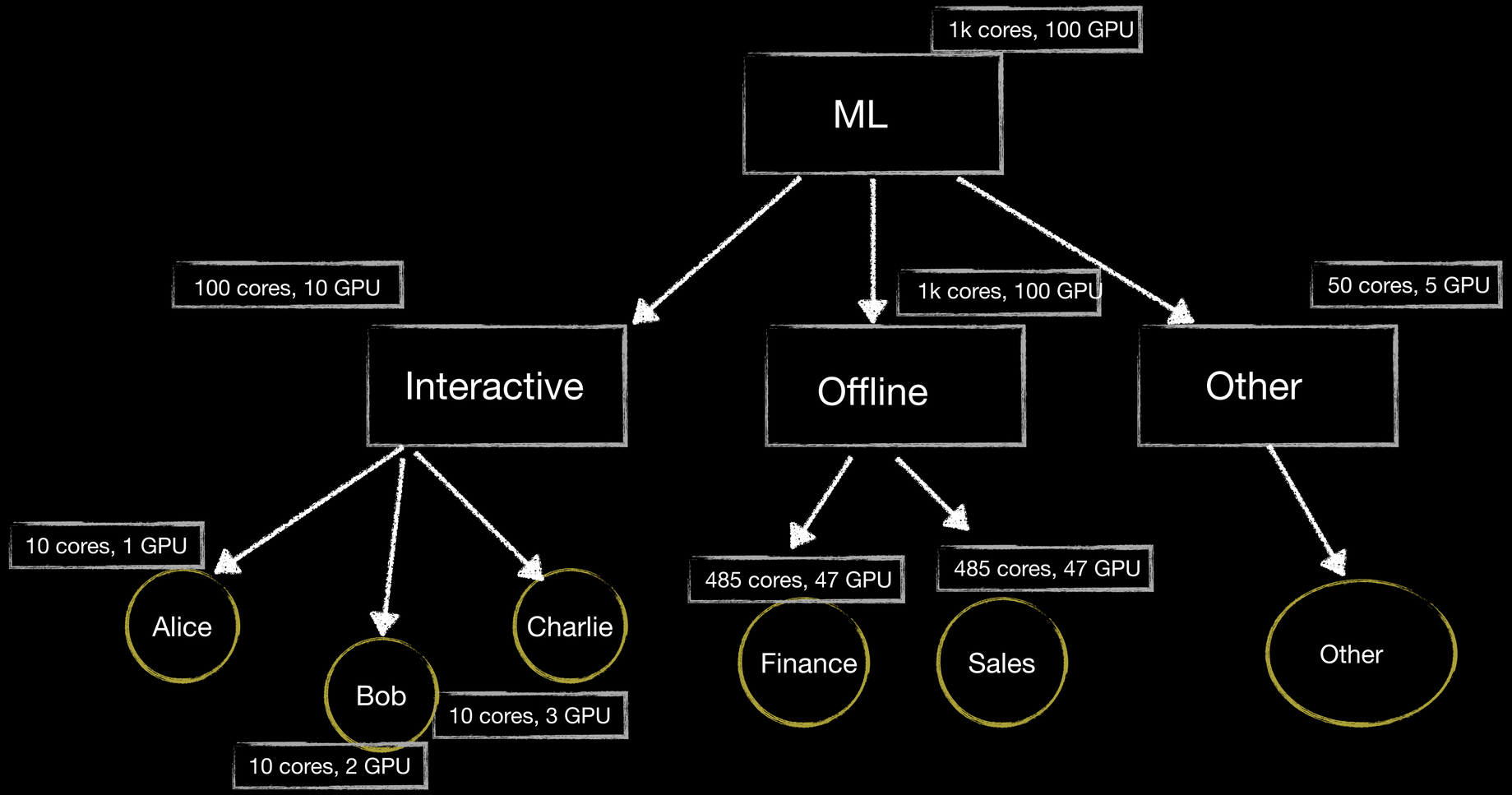


Model Training/Fine-tuning

Imagine a team of MLEs exploring models and model training. They would benefit from

- All-or-nothing Scheduling
- GPUs
- Preemption
- Autoscaling
- Guaranteed Resources

Resource Structure

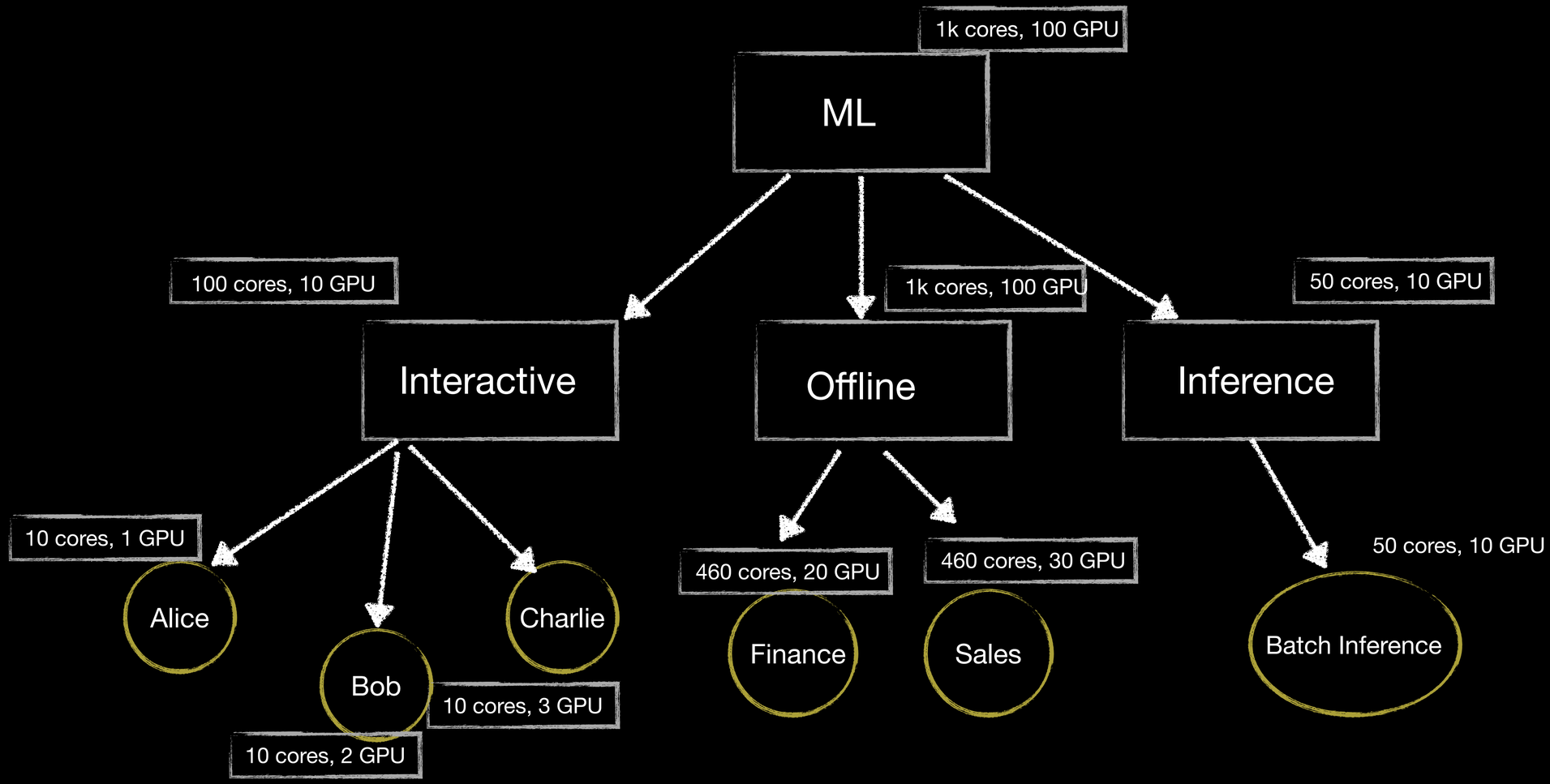


Model Inference

MLEs would benefit from

- Batch Inferencing (whenever possible)
- Preemption

Resource Structure



Summary

- Interactive development of AI apps or ML models have unique requirements, especially for large teams
  - Resource guarantees along with collaboration tools for faster iteration
  - Being able to keep the same compute stack across data and ML platforms can result in fixed cost savings
  - Scheduling and Infrastructure layers offer additional optimizations for interactive workloads

Contact

- Email: shravan.achar91@gmail.com
- LinkedIn: https://www.linkedin.com/in/shravan-achar/