Kubernetes native two-level resource management for AI workloads

Diana Arroyo and Alaa Youssef
IBM Research

October 2020

Agenda

- Why the Kubernetes scheduler is not enough
- Desired capabilities
- MCAD - open source project
- How it works
- Demo
- Call to action

- Why the Kubernetes scheduler is not enough
- Desired capabilities
- MCAD - open source project
- How it works
- Demo
- Call to action

Recent CNCF Blog Post

https://www.cncf.io/blog/2020/08/10/why-the-kubernetes-scheduler-is-not-enough-for-your-ai-workloads/

BLOG

Why the Kubernetes Scheduler Is Not Enough for Your AI Workloads

CNCF Member Blog Post

Posted on August 10, 2020

KubeCon + CloudNativeCon sponsor guest post from Alaa Youssef, manager of the Container Cloud Platform at IBM Research

Some Characteristics of AI Workloads

- Multiple concurrent learners or executors
- Co-location/affinity preferences
- Massively parallel; big number of short running tasks
- Resource hungry
- Elastic jobs
- Elapsed time may vary while total consumed resources remain the same
- Increasing trend of interactive use cases

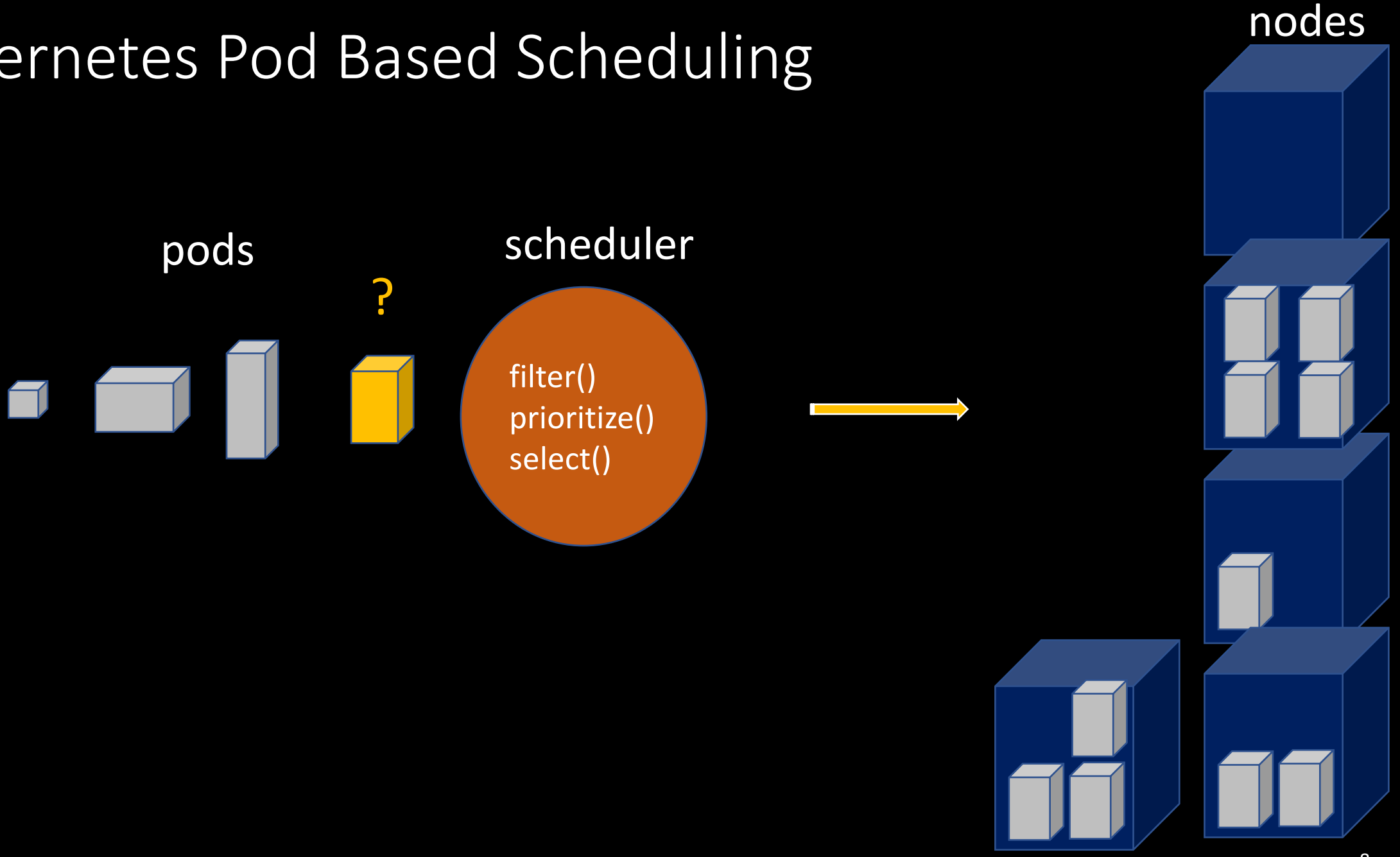
TensorFlow

Kubeflow

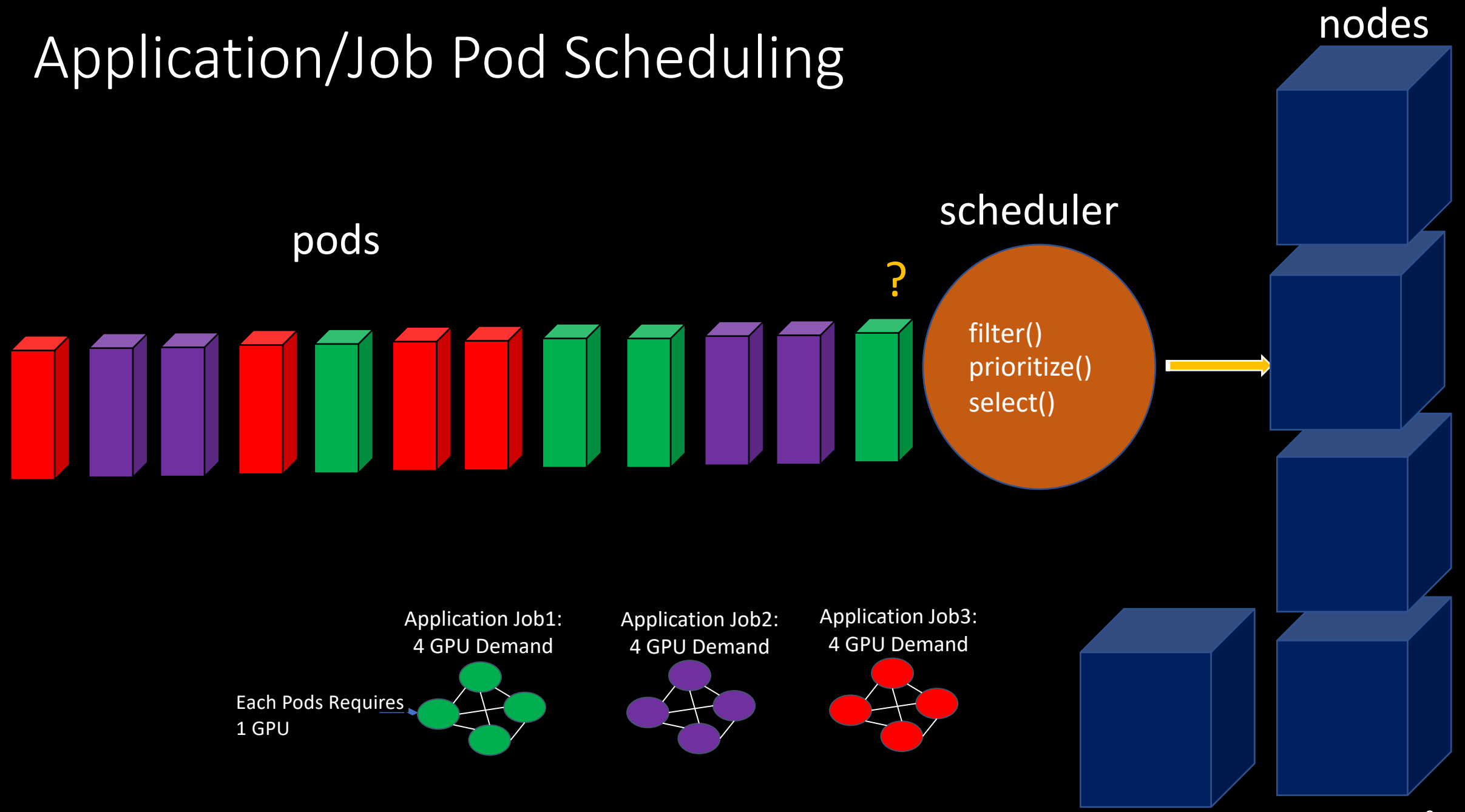
Managing Jobs vs Tasks

- Tasks map to PODs
- Jobs are composed of many tasks
- At which level do you set…
  - Priorities
  - Classes of service
  - Quotas
- Task, job, user, org, service, …?
- At which level do you queue, allocate resources, preempt, …
- What happens to your scheduler when 1000s of PODs are pending? … Overwhelming!!

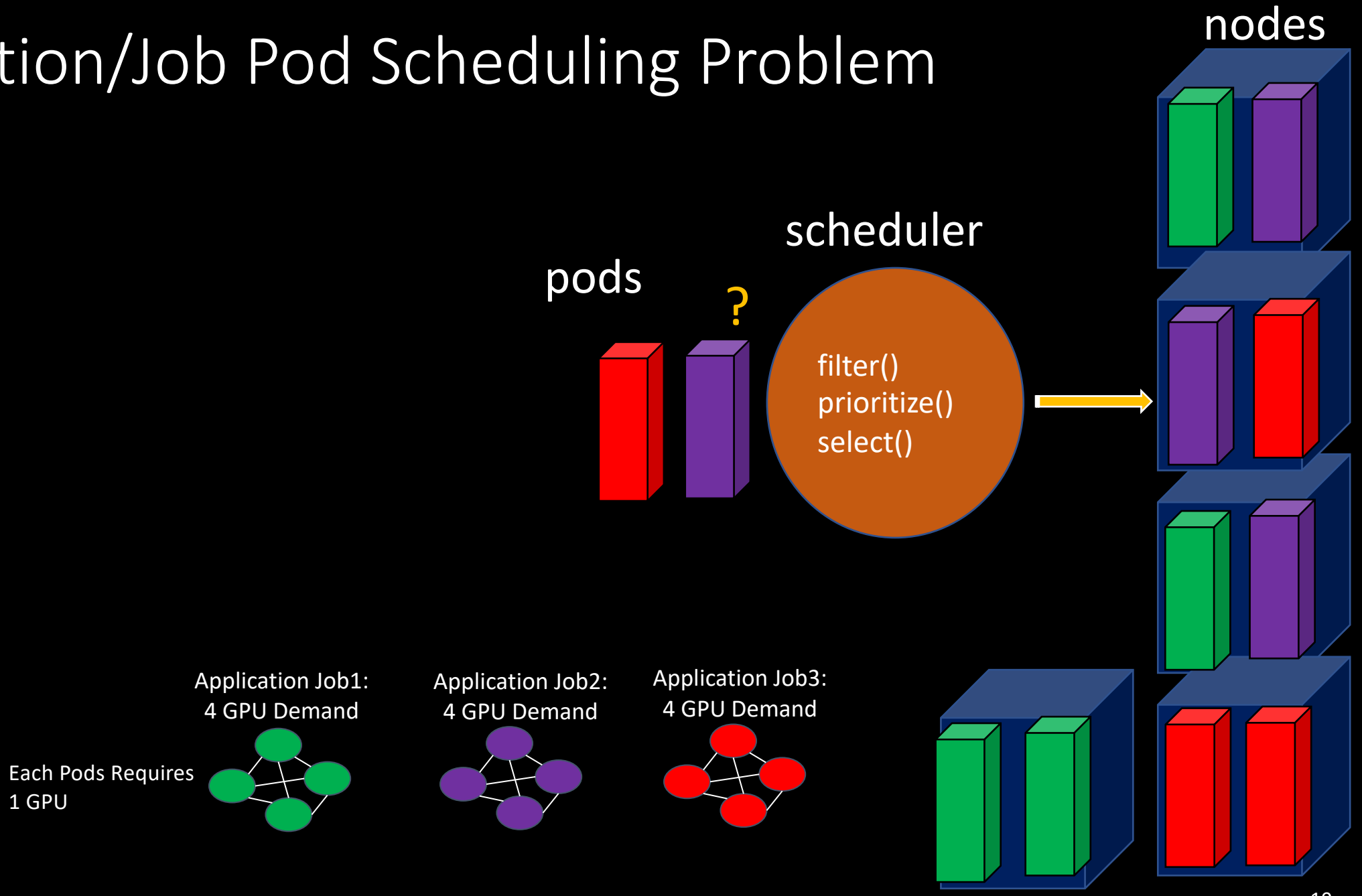
Kubernetes Pod Based Scheduling



Application/Job Pod Scheduling



Application/Job Pod Scheduling Problem



Cluster Resources Are Limited

- Scaling up the cluster by adding nodes takes time
- These jobs are resource hungry and are willing to consume any resources you provide
- Practically, there is a limit on available resources, at every moment, even if not permanent

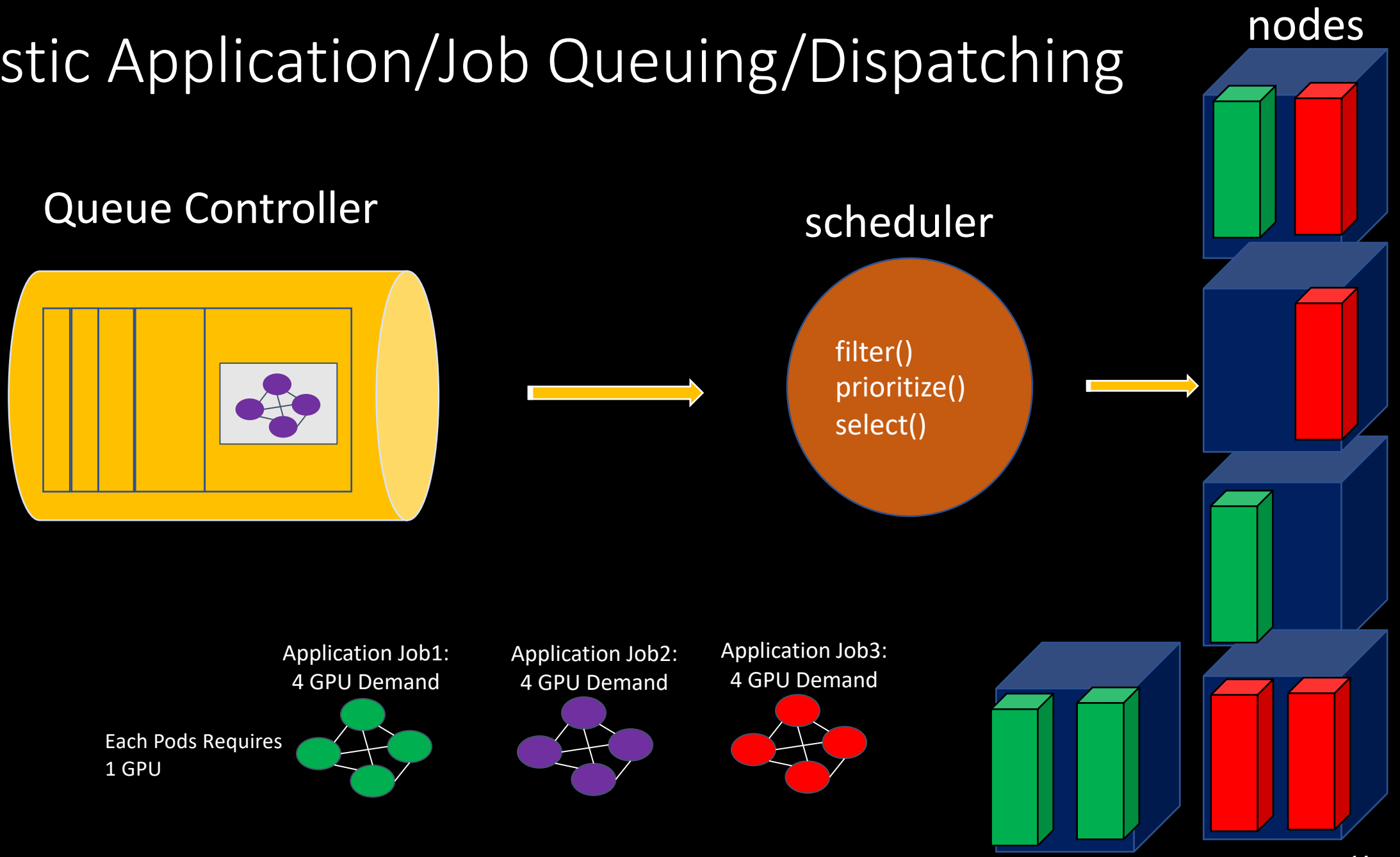
“The sky is the limit”, but this is only a cloud!

Multi-cluster, Hybrid Cloud, and Edge

- Organizations own tens of clusters
- Smaller clusters are easier to manage
- Static assignment of users or apps to clusters is not efficient
- Bursting from on-prem to public is needed more often and less predictable for AI workloads
- The rise of edge computing paradigm introduces more clusters to manage and choose from for running a job

- Why the Kubernetes scheduler is not enough
- Desired capabilities
- MCAD - open source project
- How it works
- Demo
- Call to action

Holistic Application/Job Queuing/Dispatching

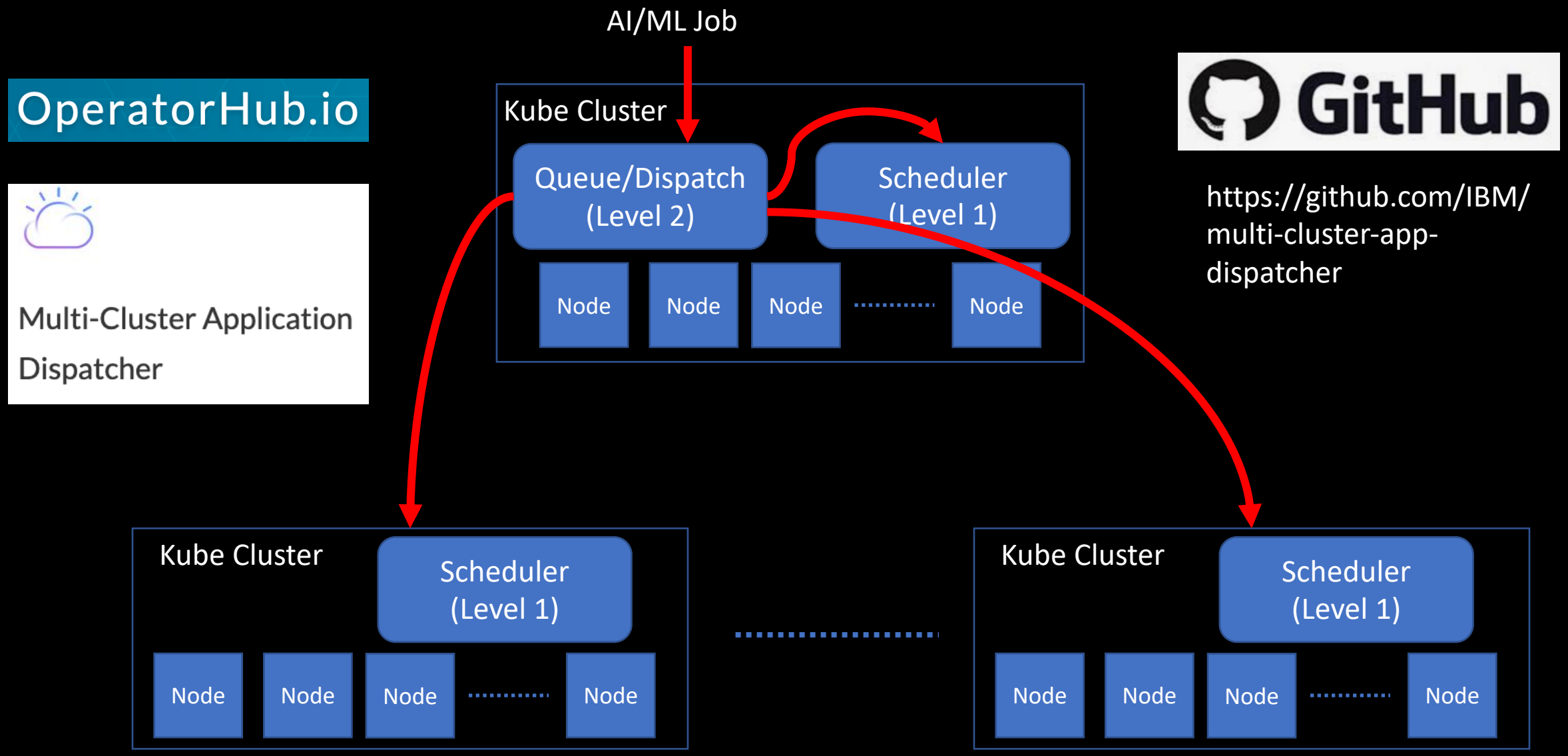


More Desired Capabilities

- Multi-cluster dispatching
- Priorities and classes of service (Gold/Silver/Bronze).
- Hierarchical quota management – soft and hard quotas, multiple resources
- Preemption
- Unified view of jobs belonging to multiple AI/ML frameworks – AppWrapper

- Why the Kubernetes scheduler is not enough
- Desired capabilities
- MCAD - open source project
- How it works
- Demo
- Call to action

What is MCAD?


AI/ML Job
OperatorHub.io
Multi-Cluster Application Dispatcher
Kube Cluster
Queue/Dispatch (Level 2)
Scheduler (Level 1)
Node
Node
Node
Node
GitHub
https://github.com/IBM/multi-cluster-app-dispatcher
Kube Cluster
Scheduler (Level 1)
Node
Node
Node
Node
Kube Cluster
Scheduler (Level 1)
Node
Node
Node
Node

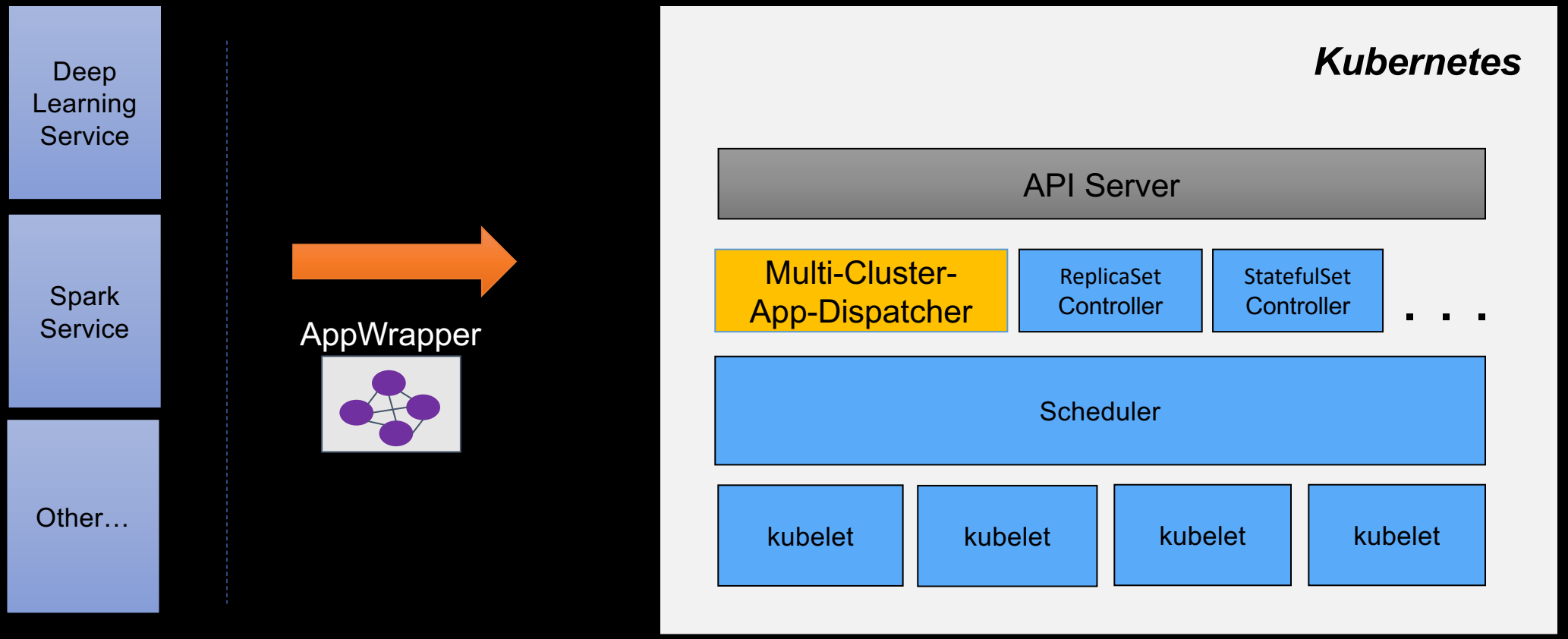
- Why the Kubernetes scheduler is not enough
- Desired capabilities
- MCAD - open source project
- How it works
- Demo
- Call to action

MCAD Controller

- AppWrapper Custom Resource Definition
  - Wraps All Compute Resource Consuming Kubernetes Objects for a Job
    - E.g. Deployments, Pods, Statefulsets
  - Wraps All Non-Compute Resource Consuming Kubernetes Objects for a Job
    - E.g. Services, Namespaces

- MCAD Custom Resource Controller
  - Determines Runnable Job for All Kubernetes Objects within an AppWrapper Holistically
  - Unwraps and Dispatches All Kubernetes Objects within an AppWrapper when Job is Runnable
  - Queues Non-Runnable Jobs
  - Supports Preemption/Requeuing

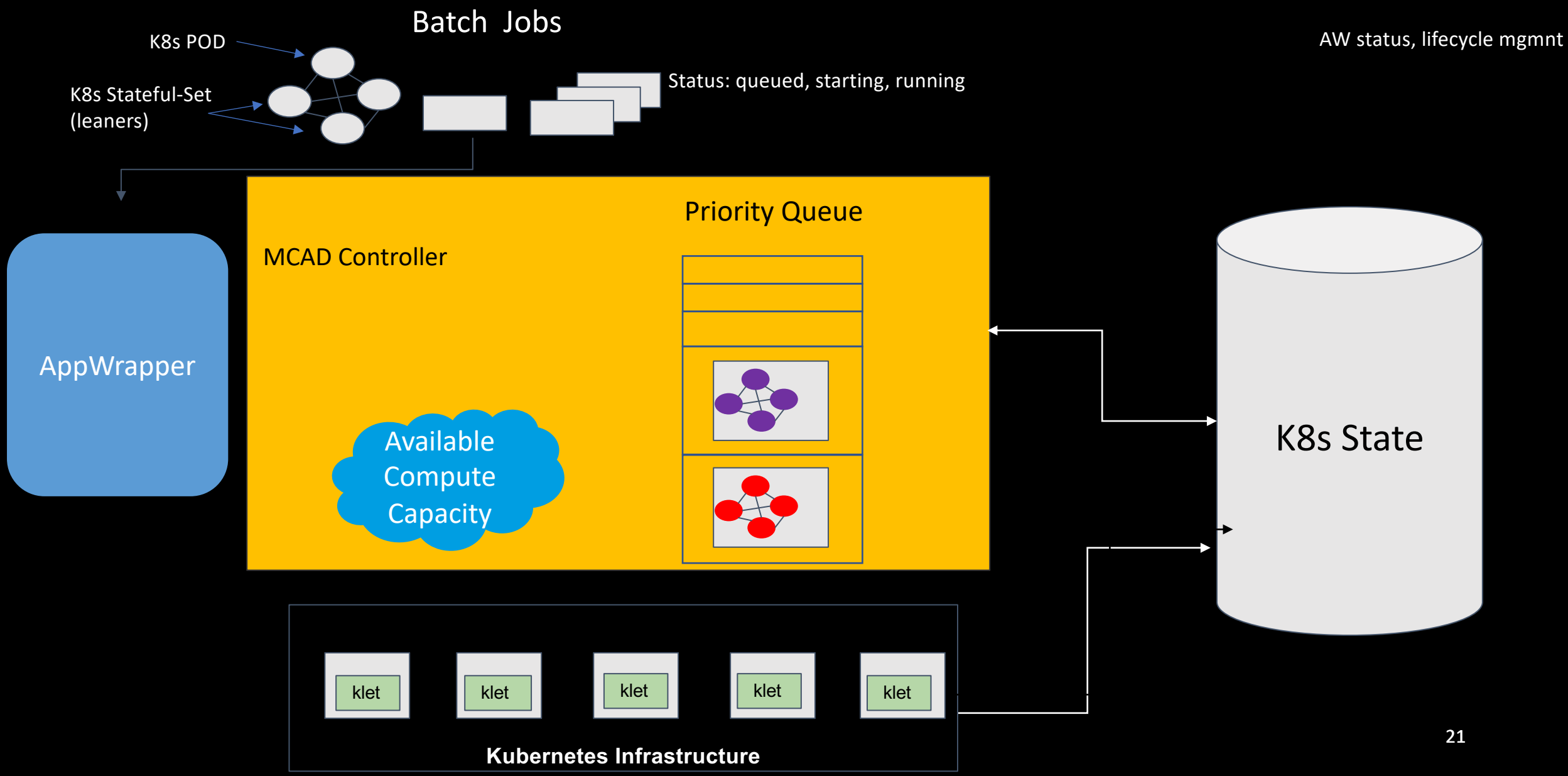
Multi-Cluster Application Dispatcher



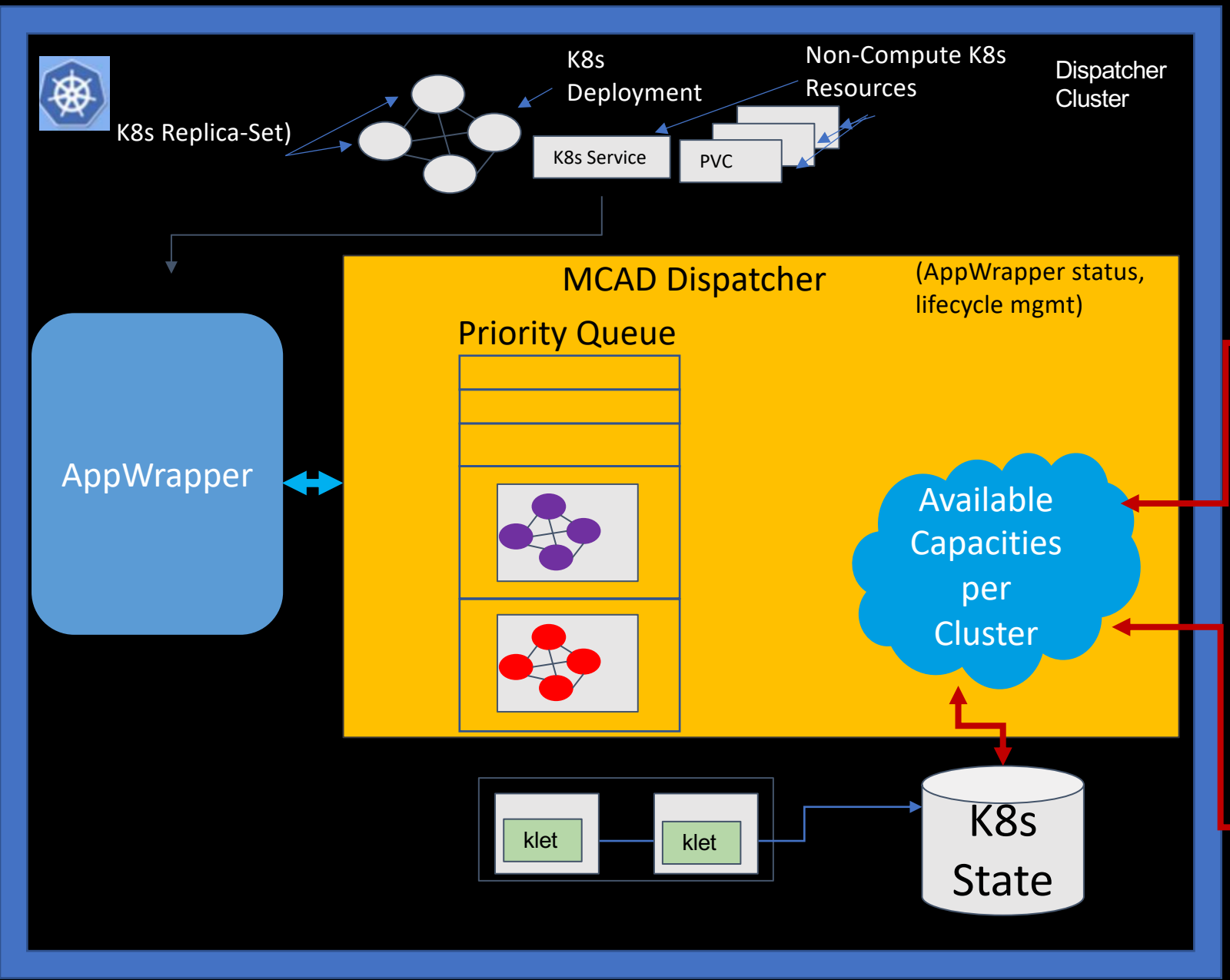
2 Runtime Configurations

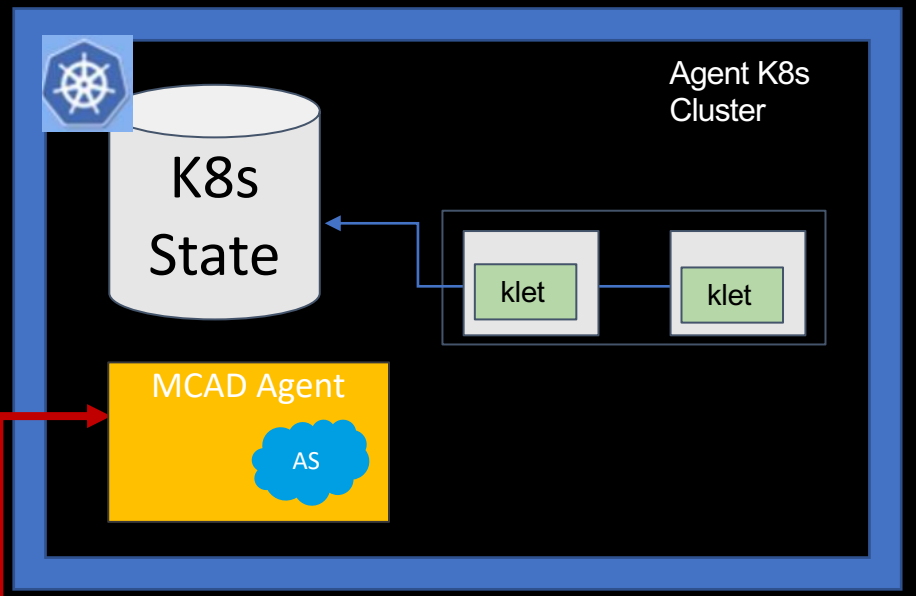
- Standalone
- Dispatcher/Agent

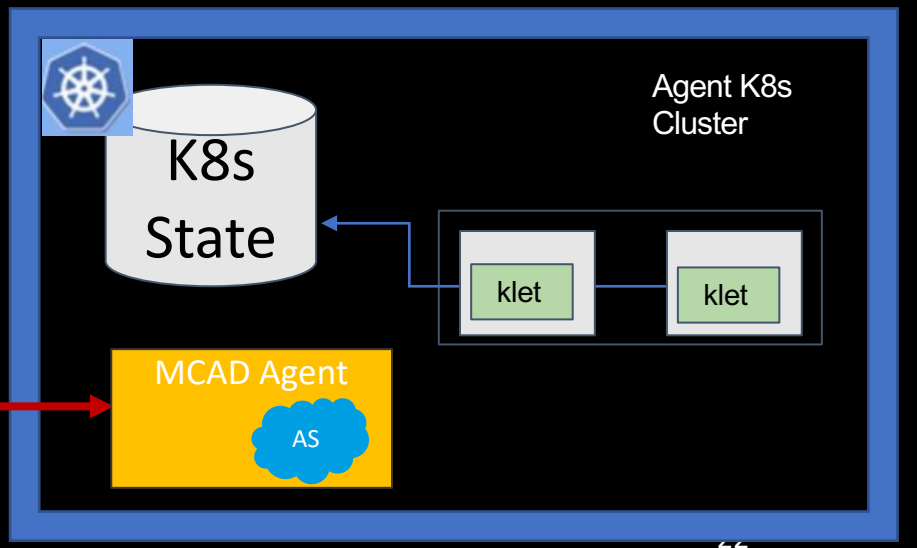
Standalone: Multi-Cluster Application Dispatcher



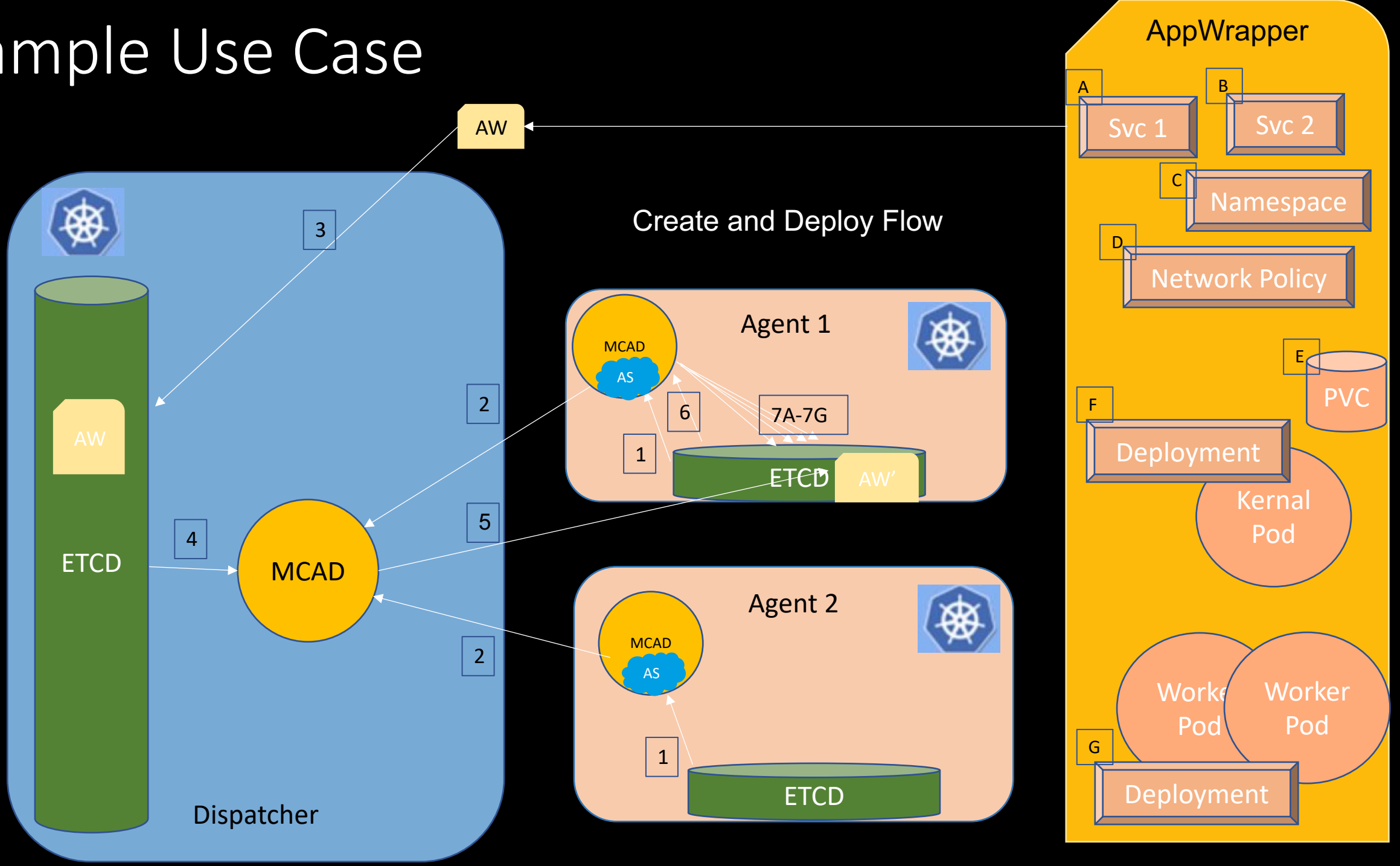
Multi-Cluster: Dispatcher/Agent



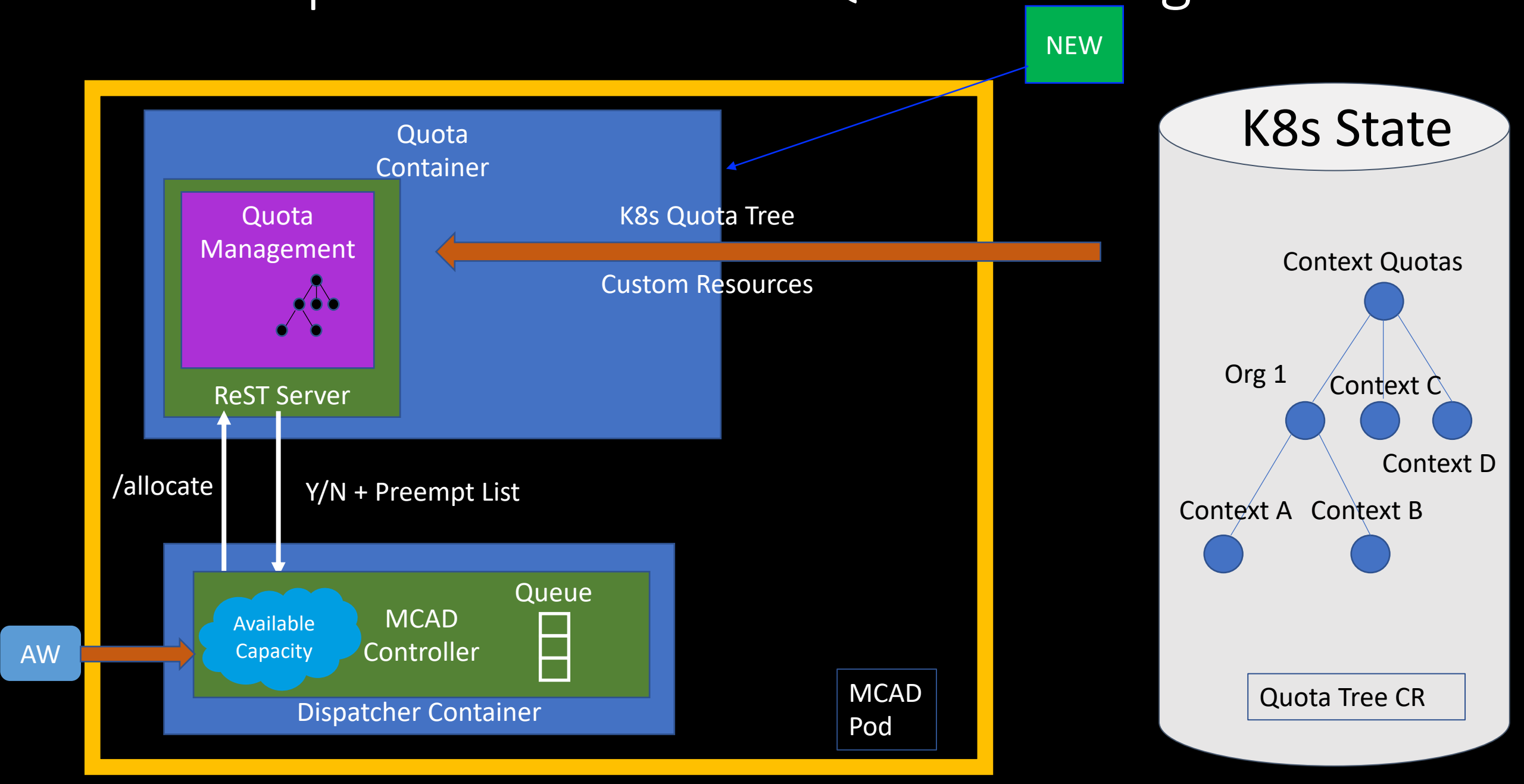




Example Use Case



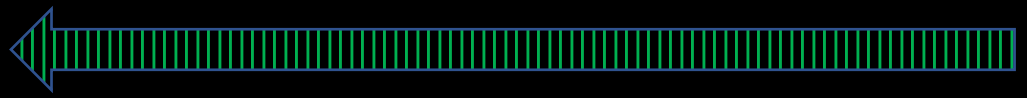
Roadmap: Current Work - Quota Management



AppWrapper Yaml

```yaml
kind: AppWrapper
metadata:
  name: job1-context2-1replica
  labels:
    quota_context: Context-2
    . . .
spec:
  priority: 1000
  resources:
    Items:
    - replicas: 1
      type: StatefulSet
      template:
        apiVersion: apps/v1
        kind: StatefulSet
        metadata:
          name: job1-context2-1replica
          labels:
            app: job1-context2-1replica
```

AppWrapper Custom Resource



(Roadmap) Quota Management

(Optional) Dispatch Priority

List of Kubernetes Resources

- Why the Kubernetes scheduler is not enough
- Desired capabilities
- MCAD - open source project
- How it works
- Demo
- Call to action

- Why the Kubernetes scheduler is not enough
- Desired capabilities
- MCAD - open source project
- How it works
- Demo
- Call to action

We Need Your Help!!

OperatorHub.io

Multi-Cluster Application Dispatcher

Try it out!
Give feedback!
Contribute!

GitHub

https://github.com/IBM/multi-cluster-app-dispatcher