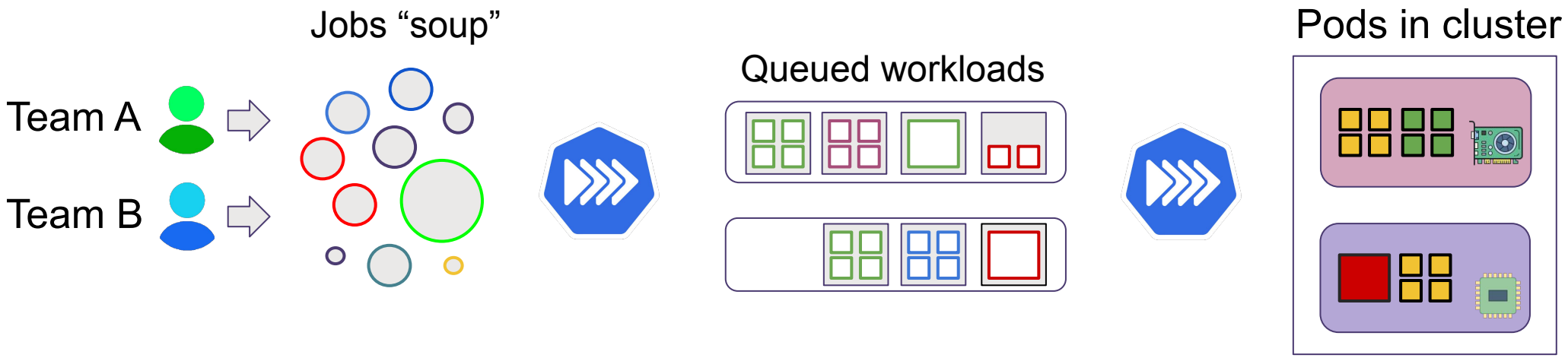
Advanced Resource Management for Running AI/ML Workloads with Kueue

Michał Woźniak (@mimowo), Google
Yuki Iwai (@tenzen-y), CyberAgent, Inc.

What is Kueue?

Job-level scheduler

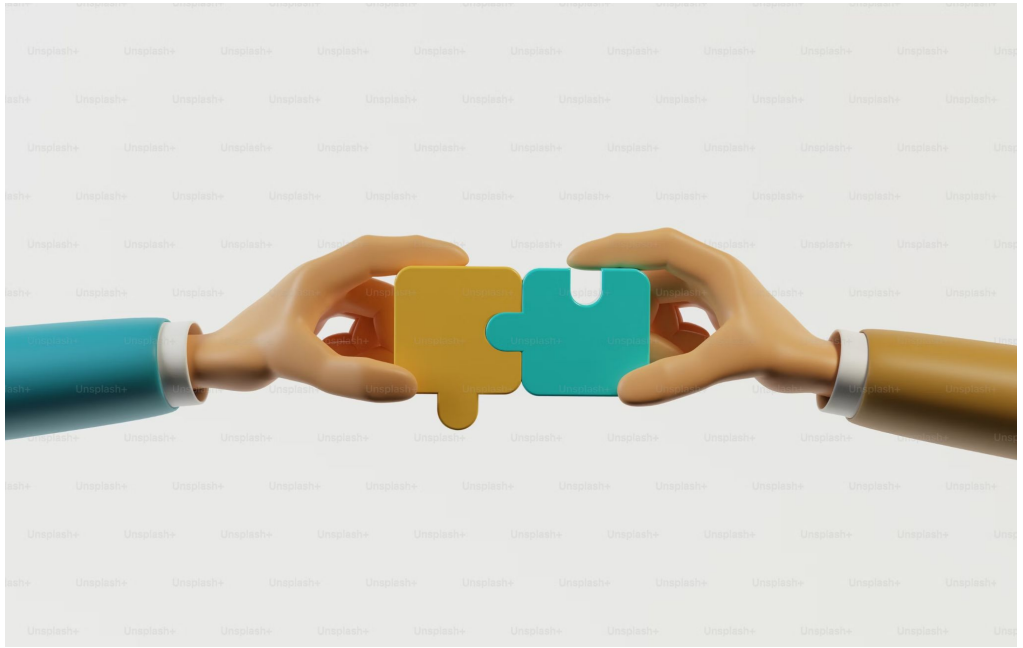
- Determines when to start a job
  - delayed pod creation off-loads api-server and kube-scheduler
- Support all-or-nothing semantics (without it jobs could “deadlock”)
- Resource quota management
  - Multi-tenant quota management
  - Quota management for various types of hardware
  - Control over resource preferences (reservations vs. on-demand vs spot)



Kueue - Design principles

It is “kube-native”:

- There is no external database
- Compatible with kube components:
  - cluster-autoscaler, kube-scheduler, Job controller
  - No kube components are forked or replaced
- Required enhancements are requested upstream



source

Kueue - Design principles

- Pods creation is gated by job admission
  - Introduce “suspend” semantics across Job CRDs

- Supported Job-based integrations:
  - k8s: Job, JobSet
  - Kubeflow Jobs
  - Ray: RayJob, RayCluster
  - In-house Jobs via Job framework

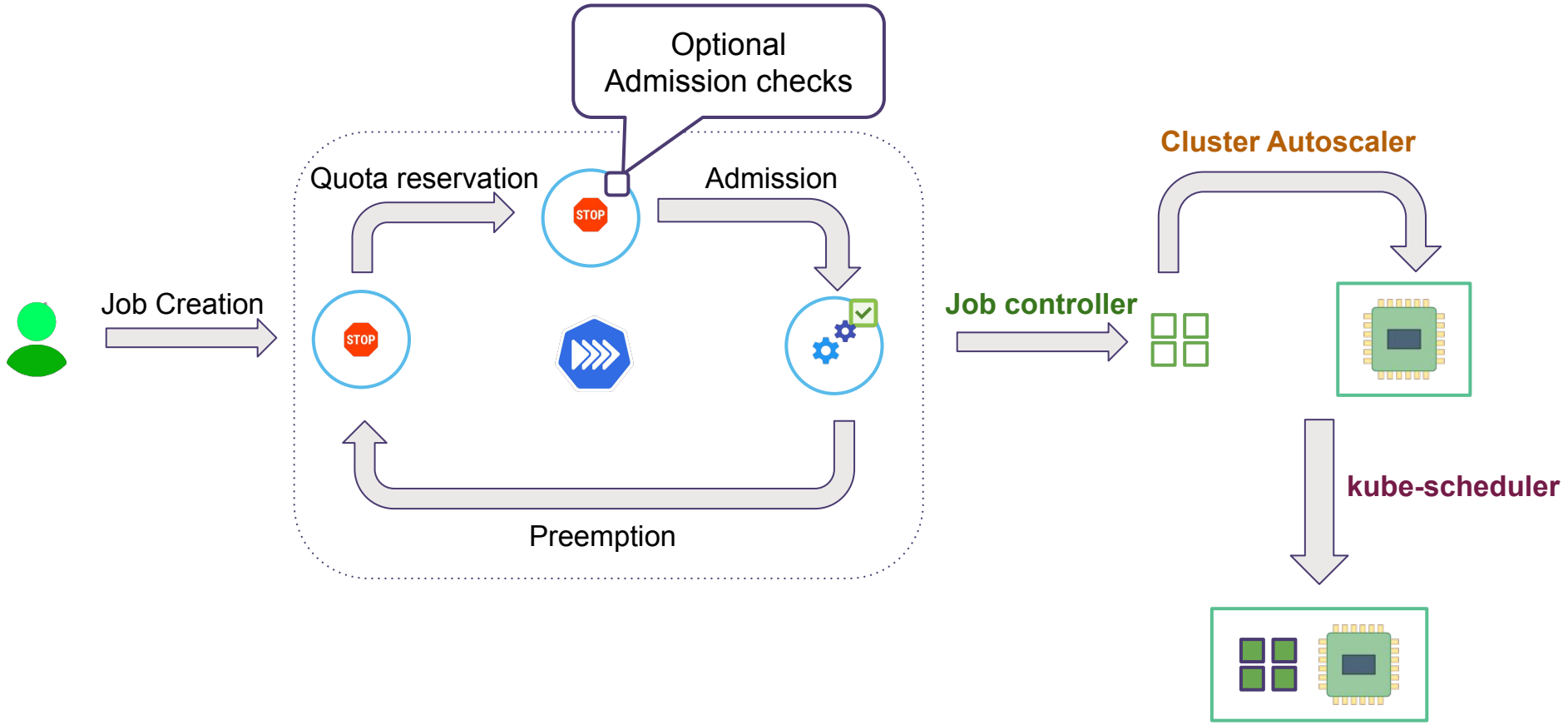
BTW: Plain pod groups are also supported

Kubeflow

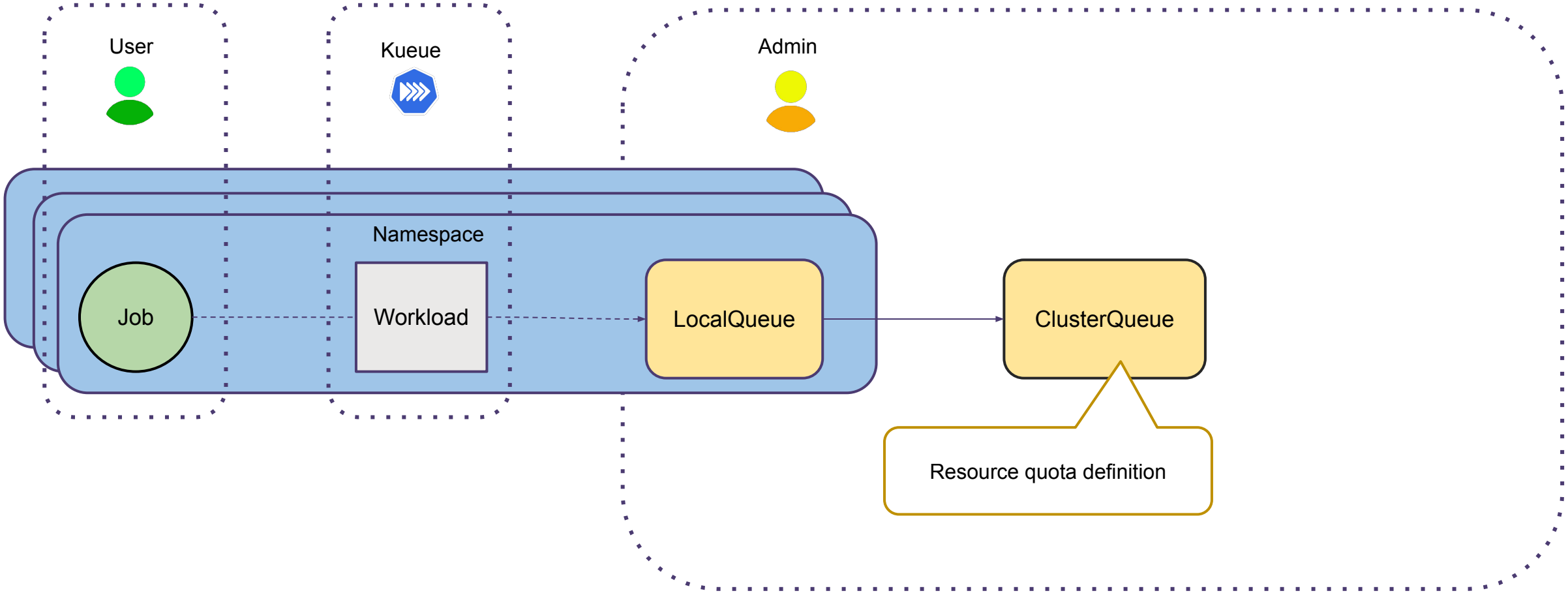
JobSet

TensorFlow

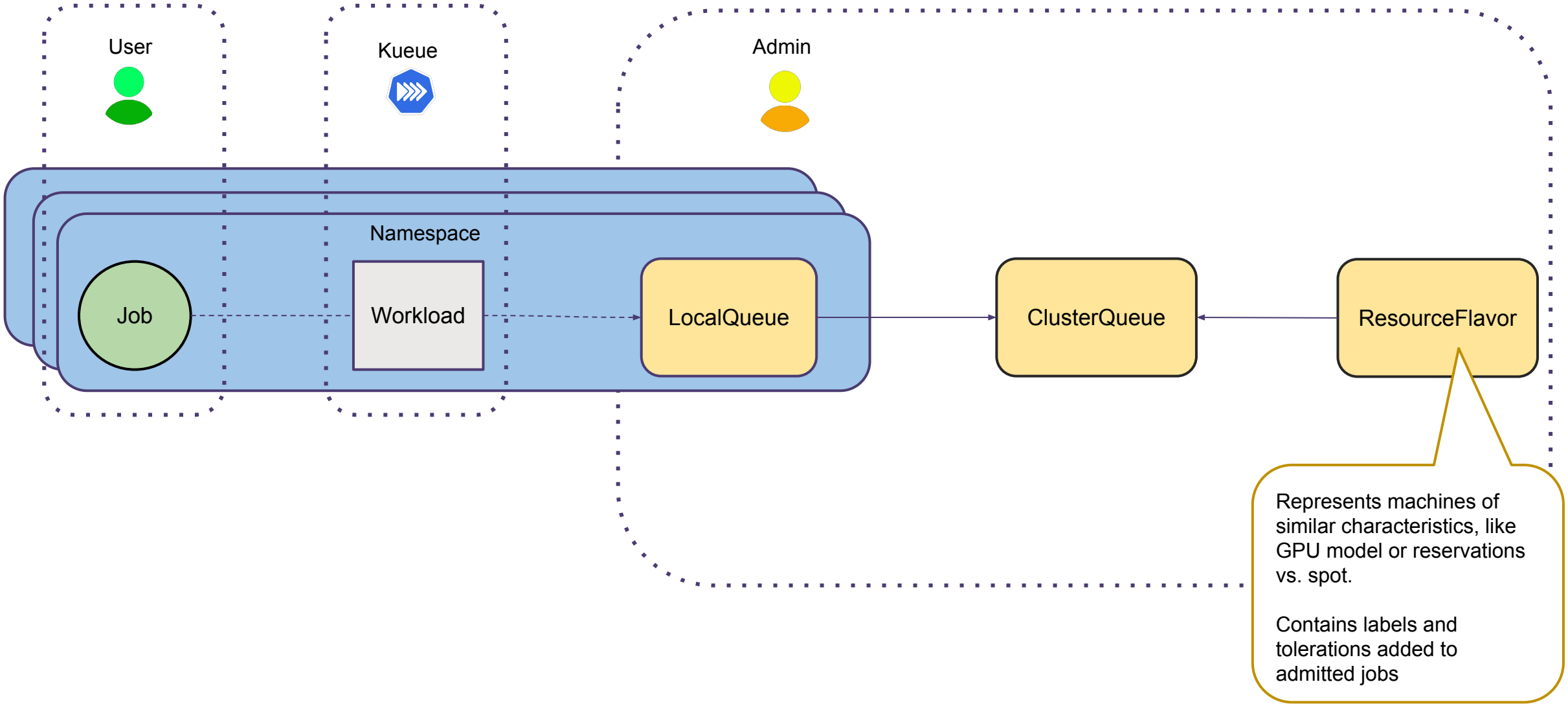
Kueue - Job lifecycle



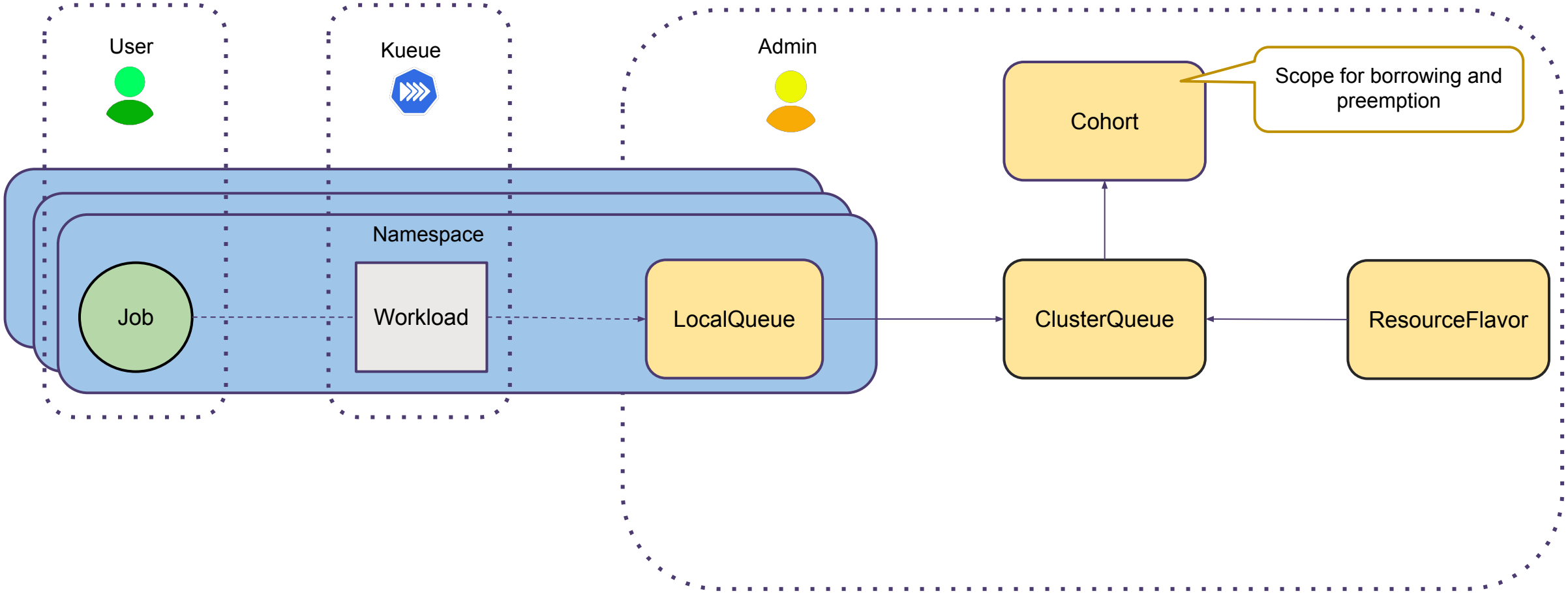
Kueue - main concepts



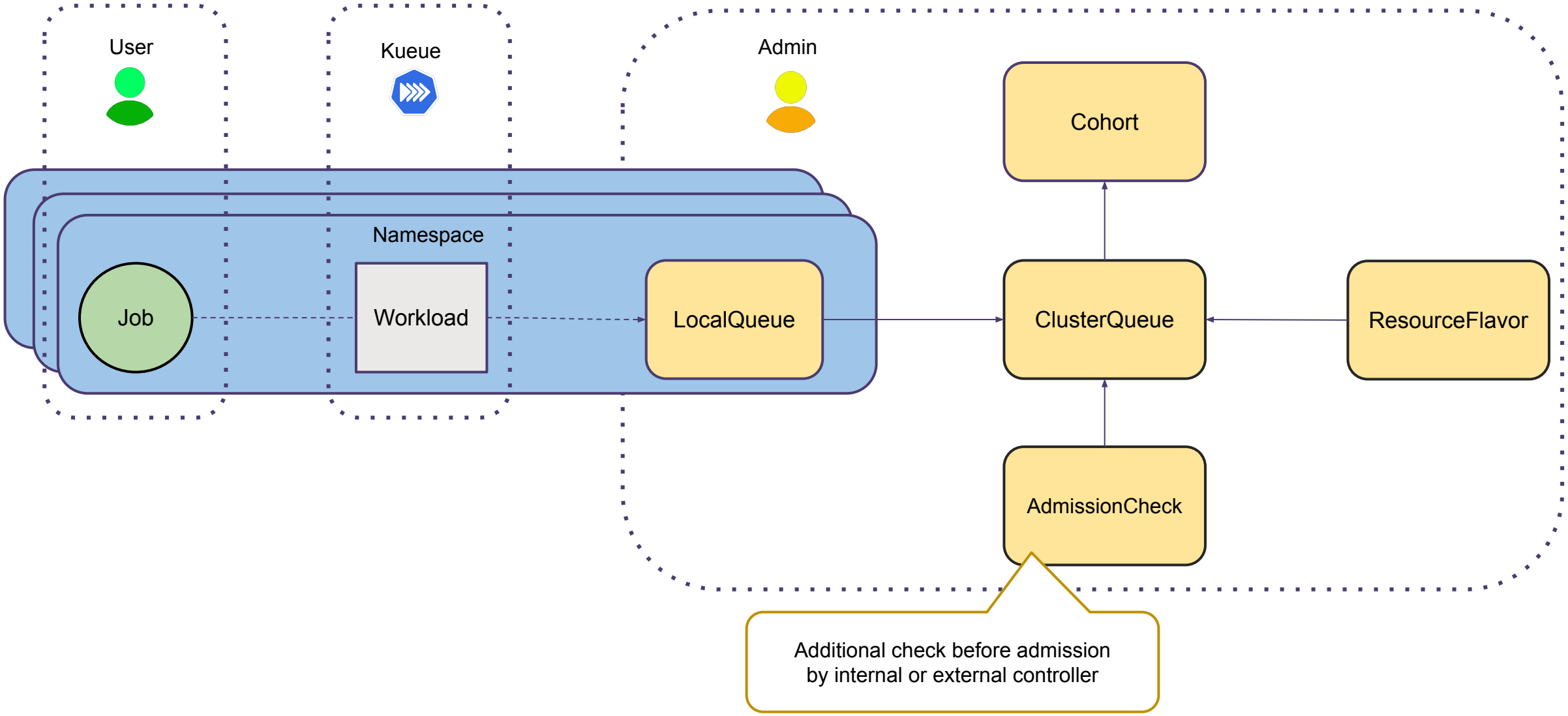
Kueue - main concepts



Kueue - main concepts



Kueue - main concepts



Batch reference architecture

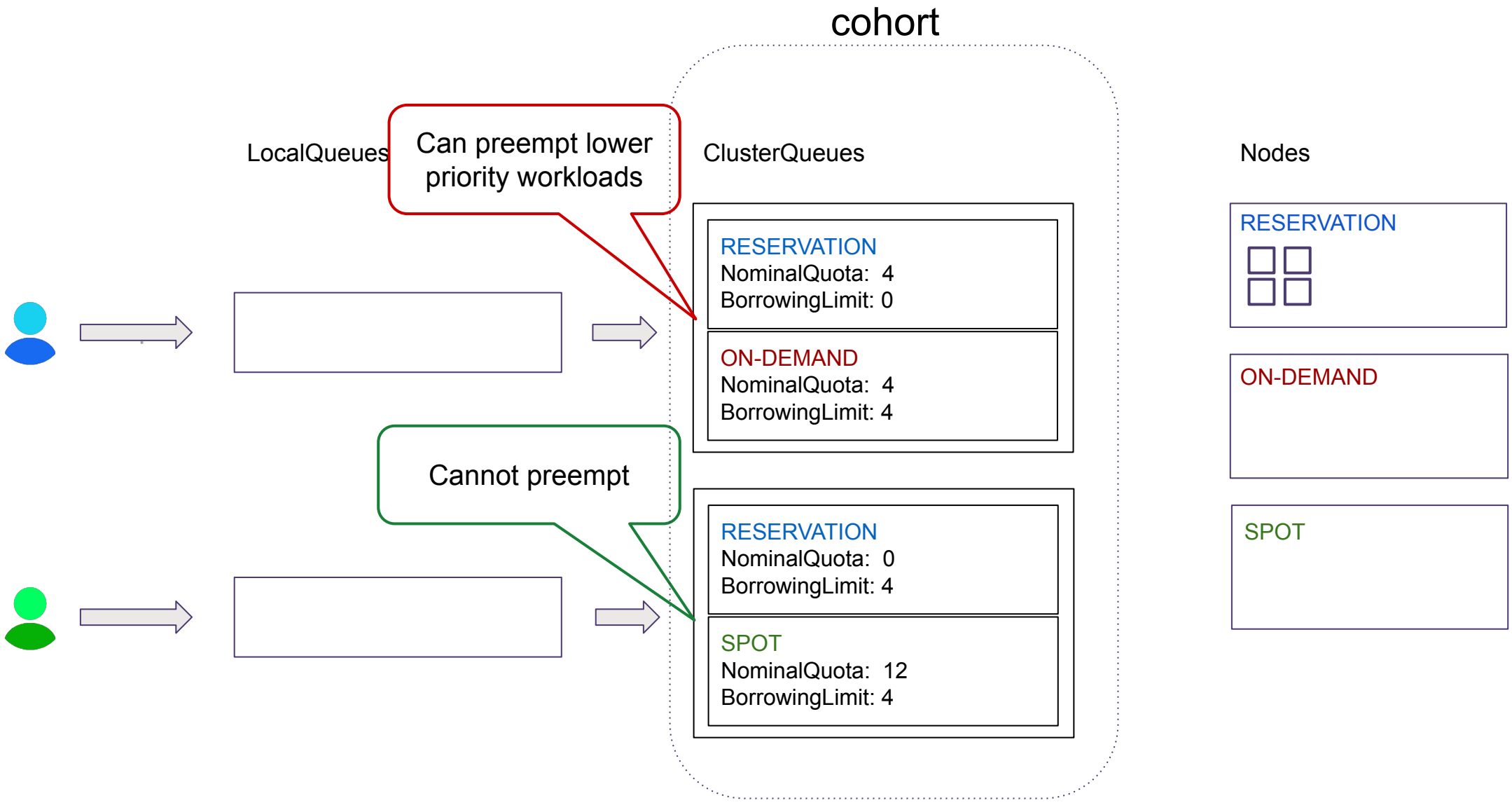
Goals:

- Intended to help platform administrators, cloud architects, and operations professionals deploy a batch processing platform
- Best practices for running batch workloads
- Customizable for user preferences

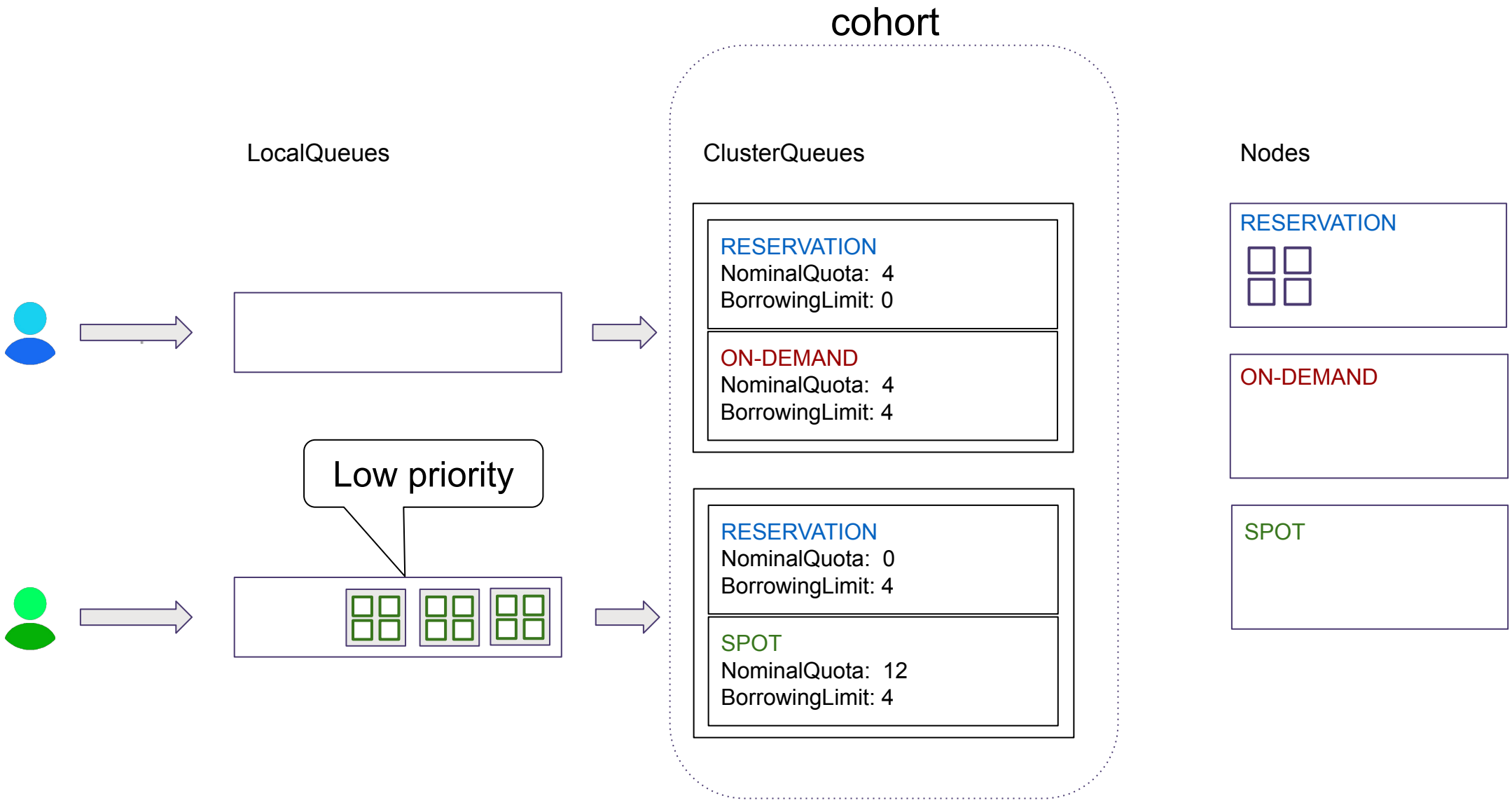
https://github.com/GoogleCloudPlatform/ai-on-gke/tree/main/gke-batch-refarch

Acknowledgements: Ali Zaidi

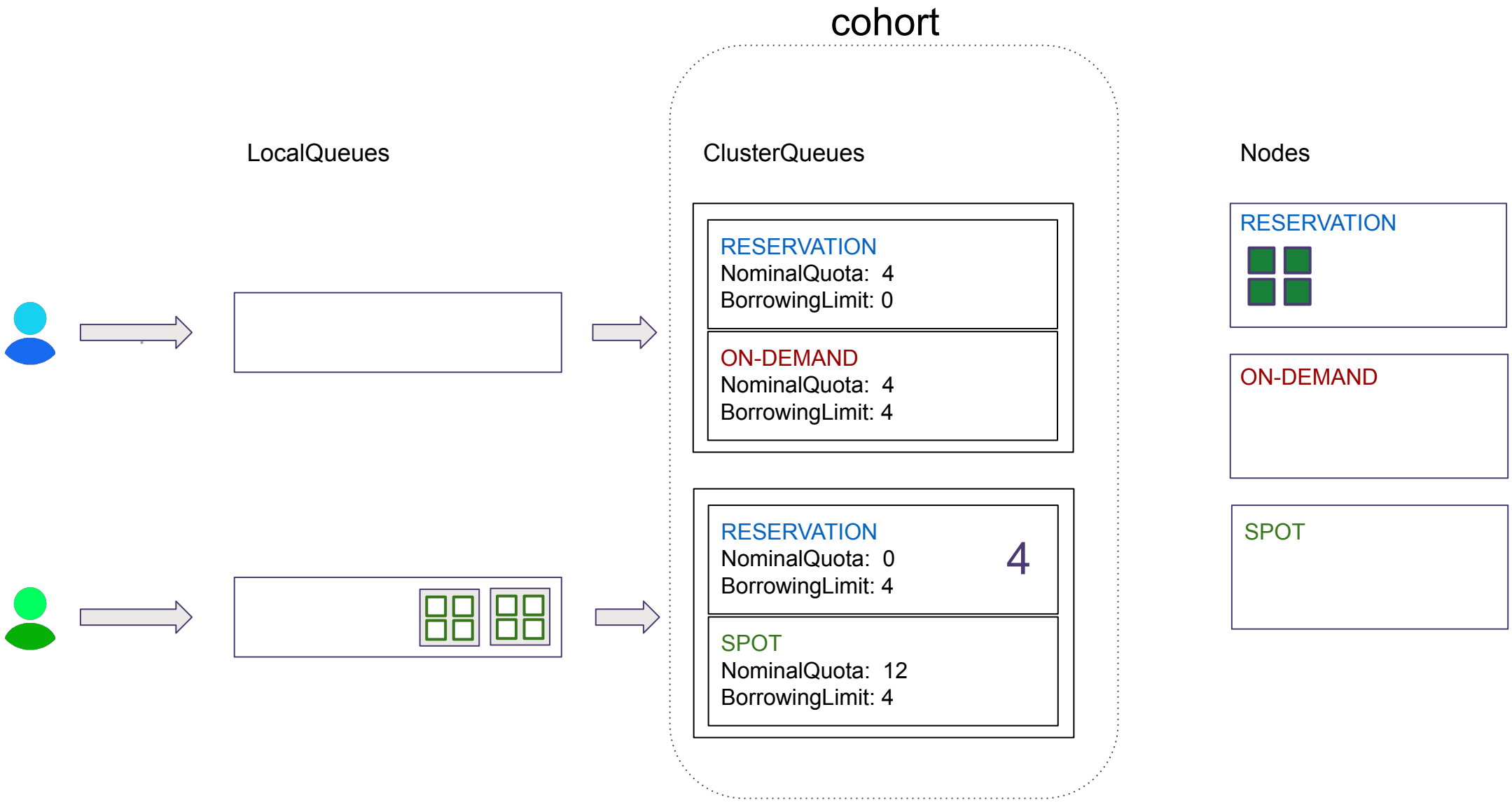
Batch reference architecture



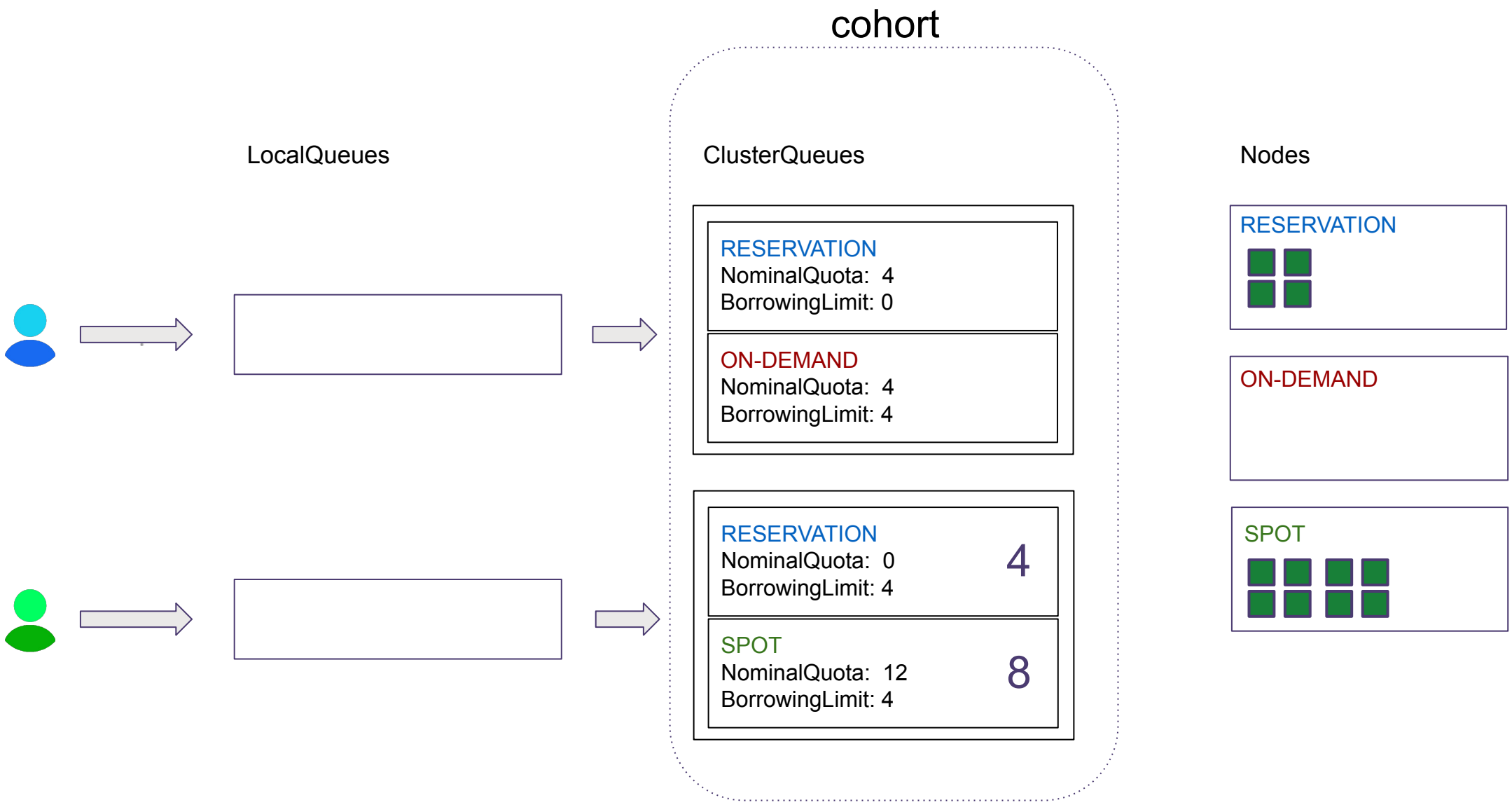
Batch reference architecture



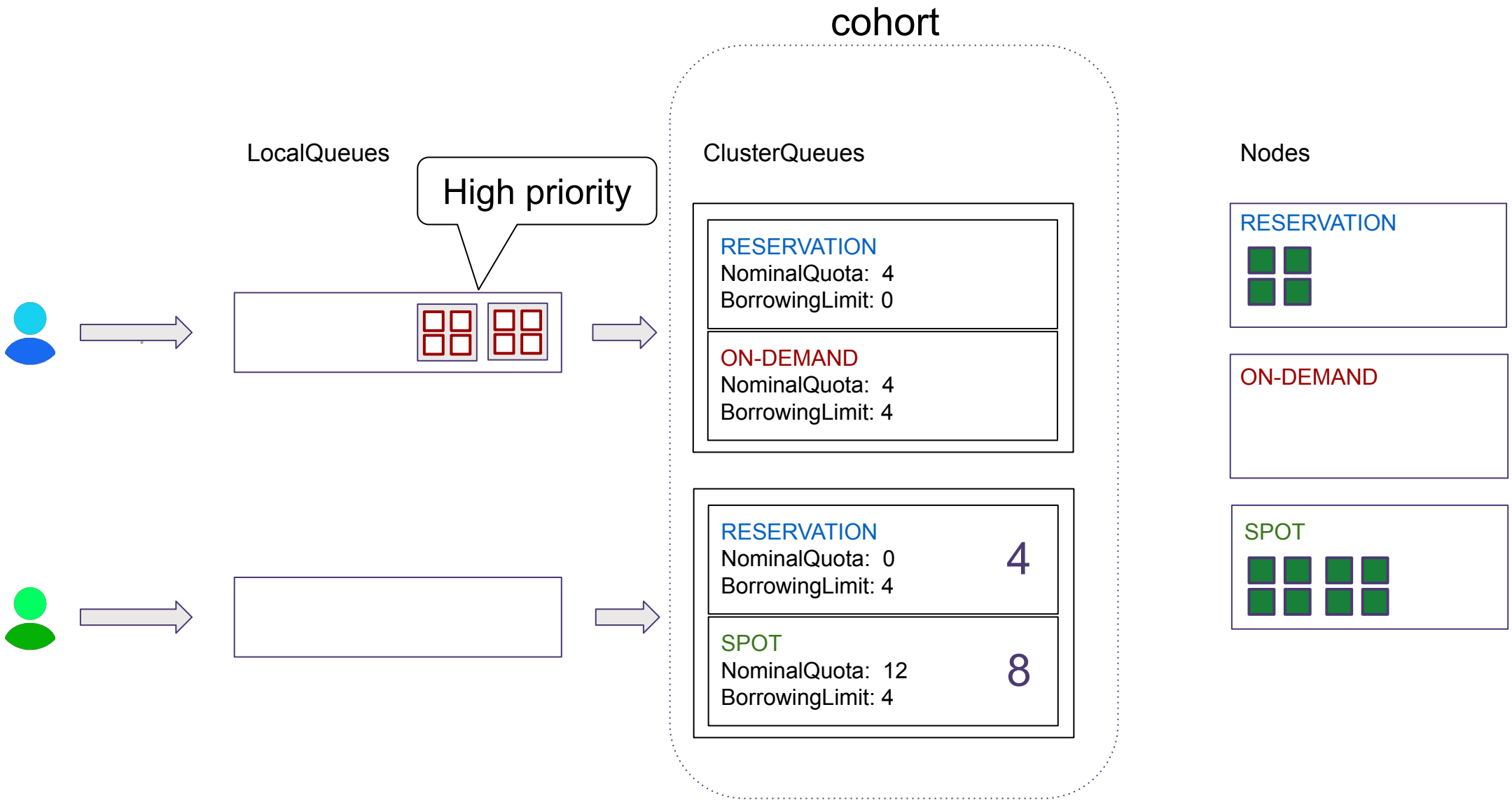
Batch reference architecture



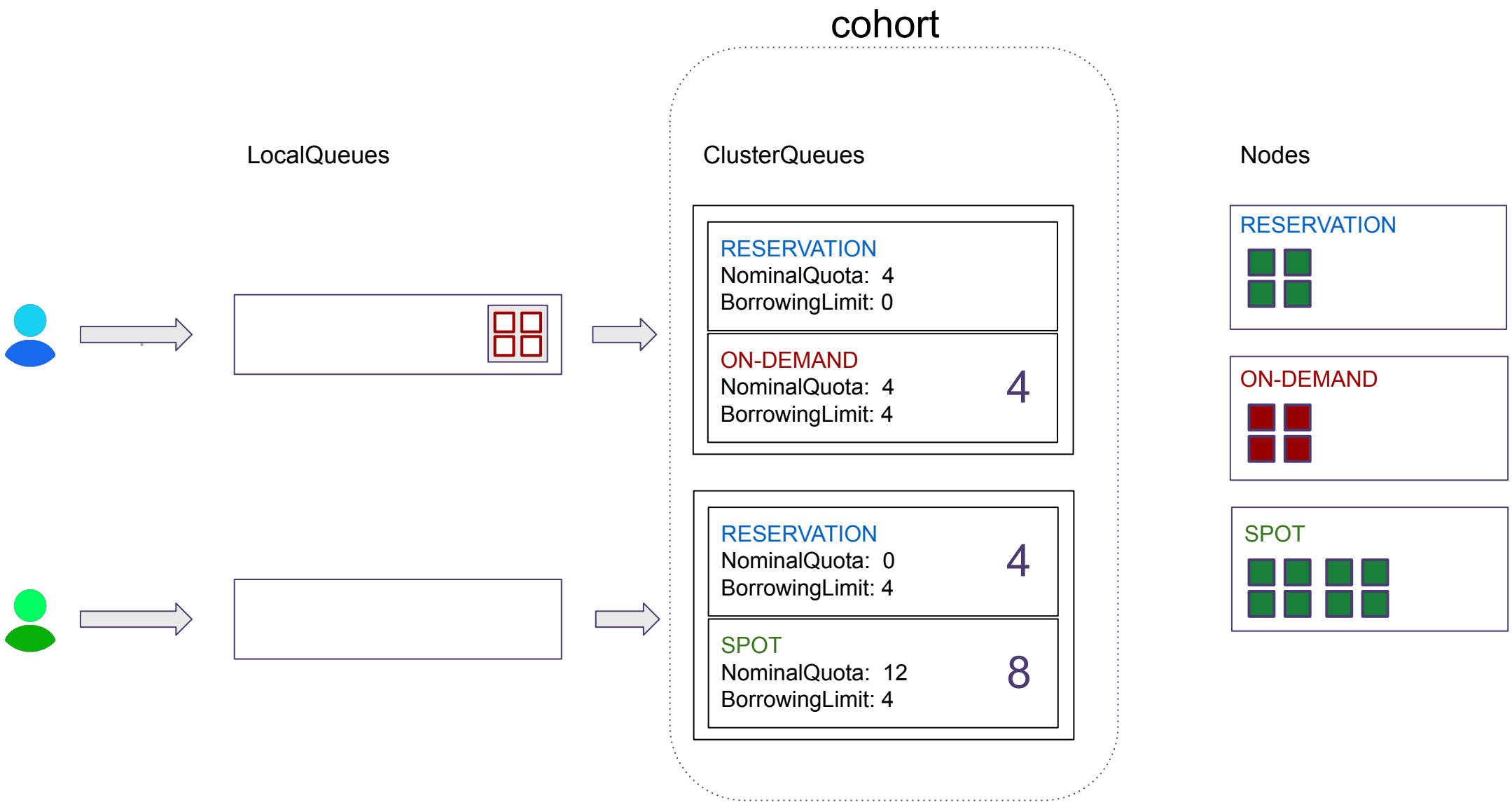
Batch reference architecture



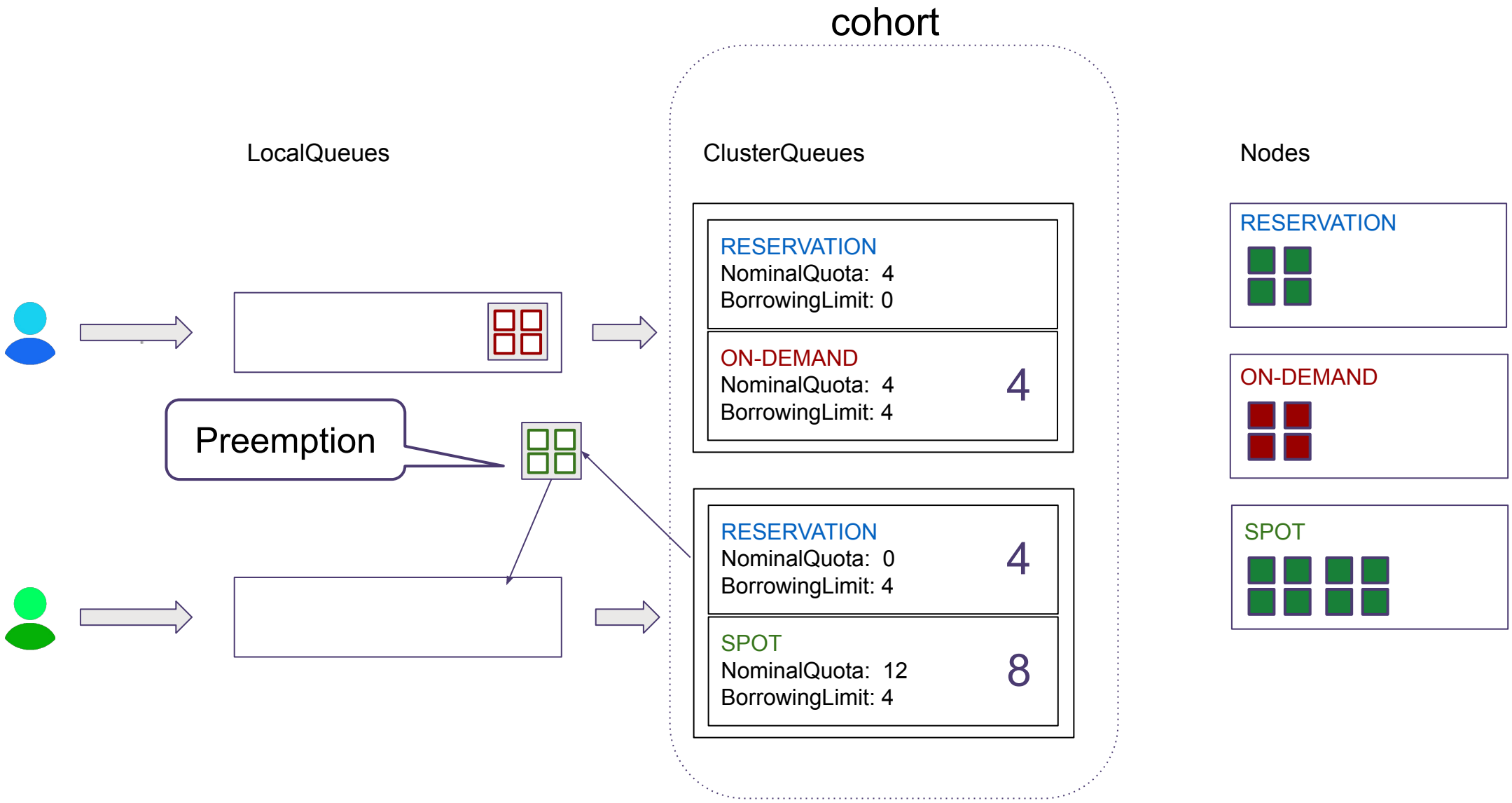
Batch reference architecture



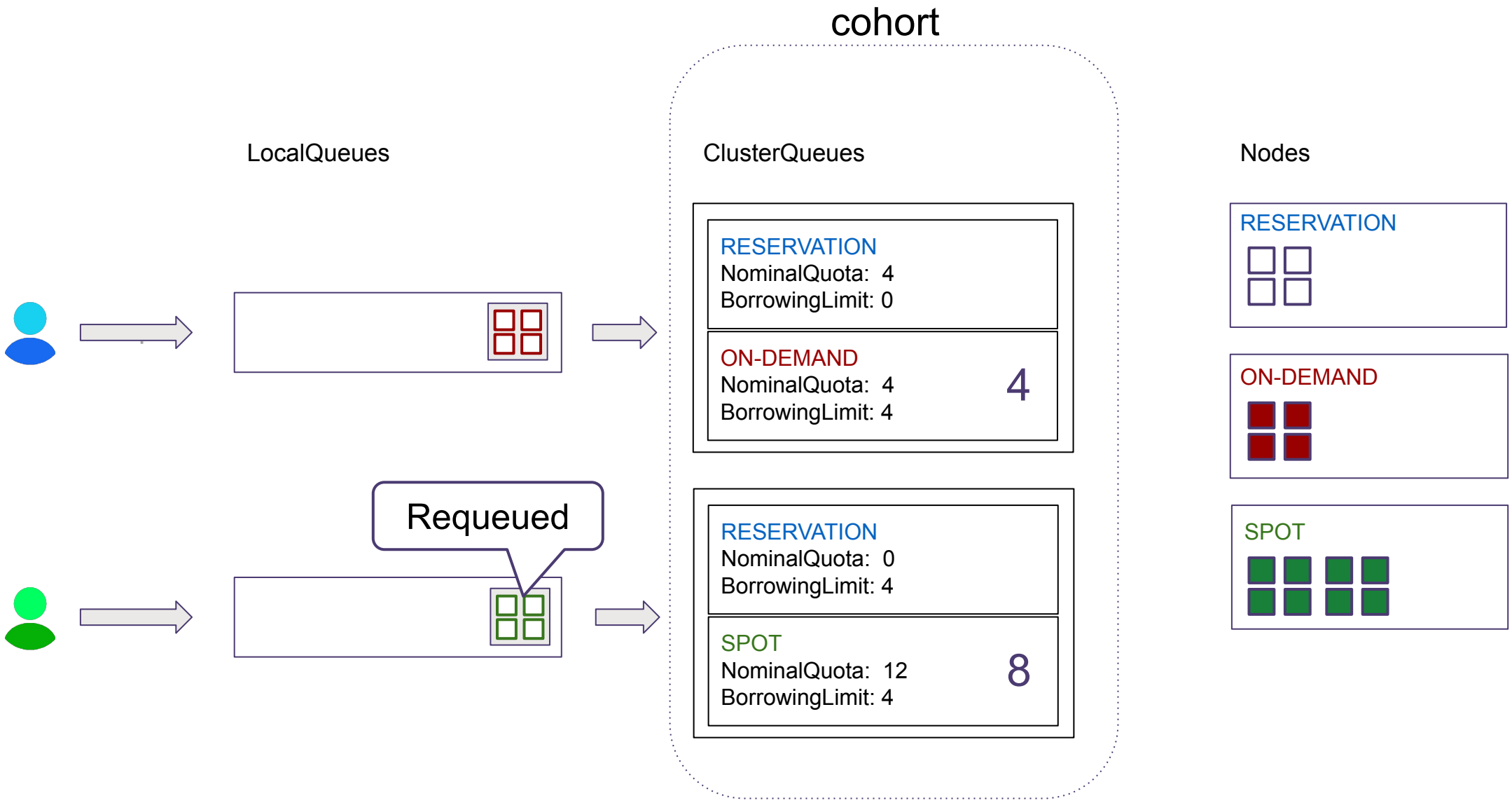
Batch reference architecture



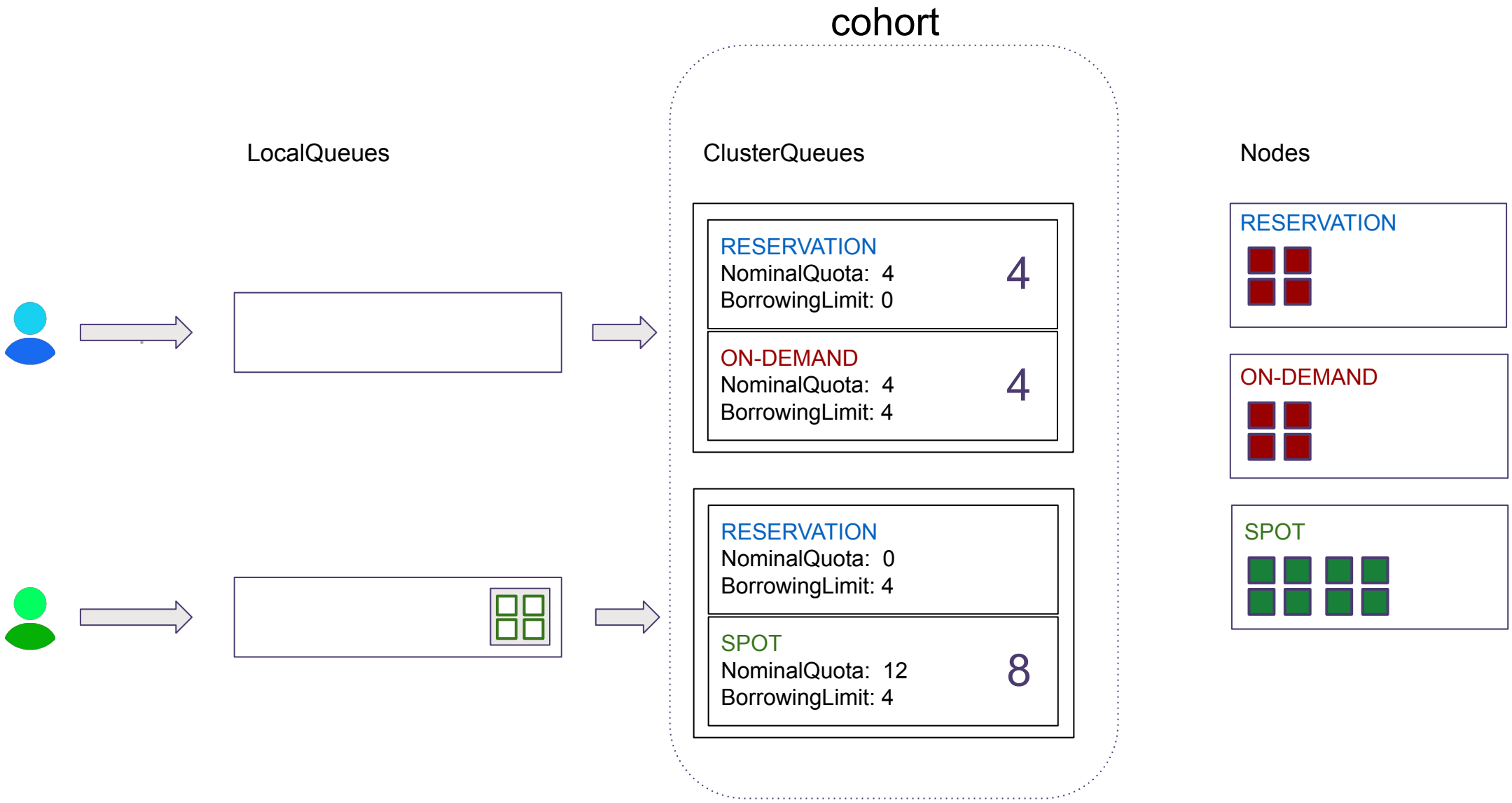
Batch reference architecture



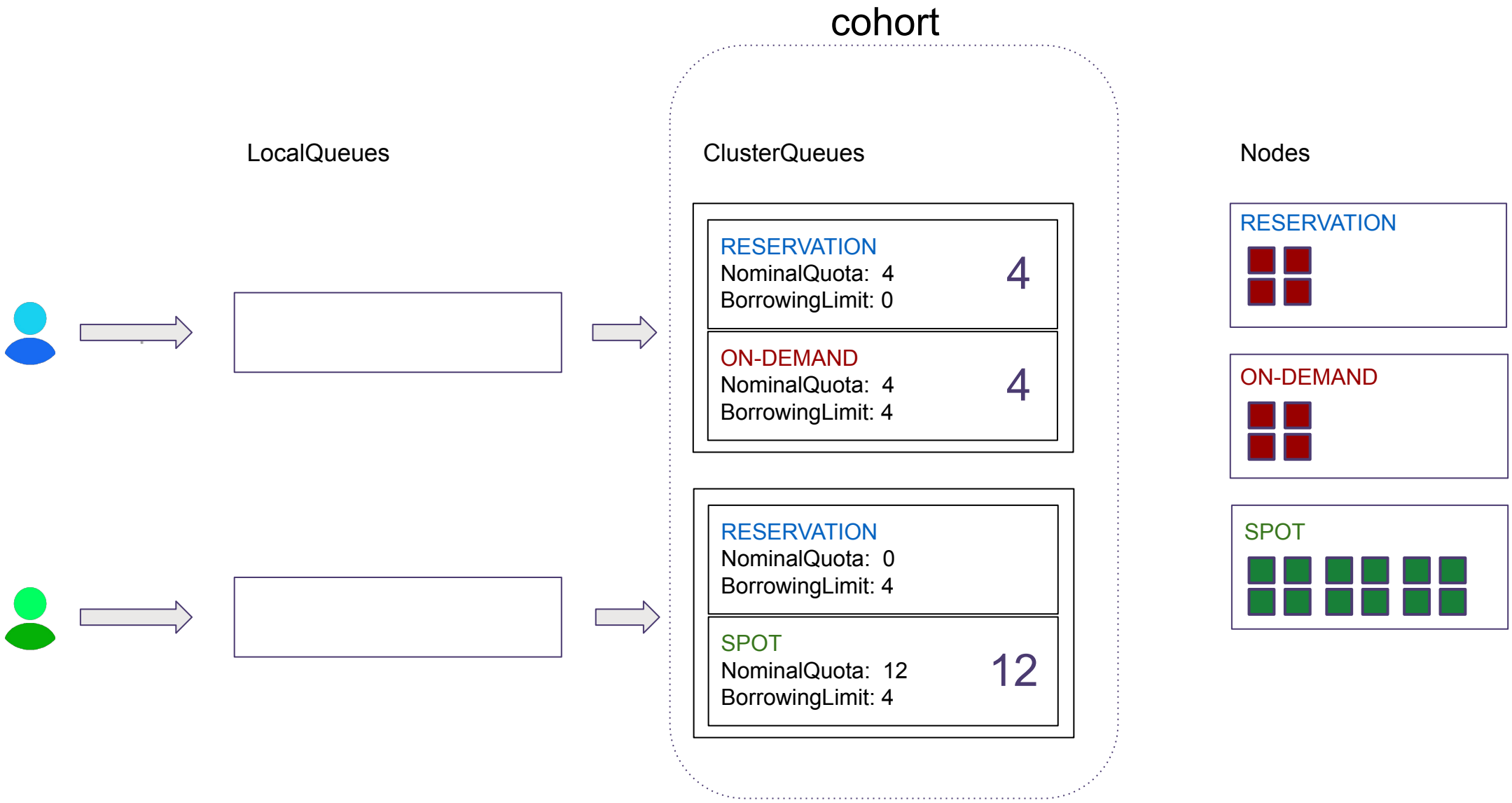
Batch reference architecture



Batch reference architecture



Batch reference architecture



Kueue in CyberAgent

source

Kueue in CyberAgent

CyberAgent Internal on-premise ML Platform

Infrastructure

- Bare metal machines for GPU Nodes
- Heterogeneous computing resources with 7 types GPUs
  - NVIDIA H100
  - NVIDIA A100 40GB
  - NVIDIA A100 80GB
  - NVIDIA L4
  - etc …

Kubernetes Cluster

- A Single Vanilla Multi-Tenant Cluster
- The number of tenants is over 300
- Operation Period is over 4 years in the same cluster

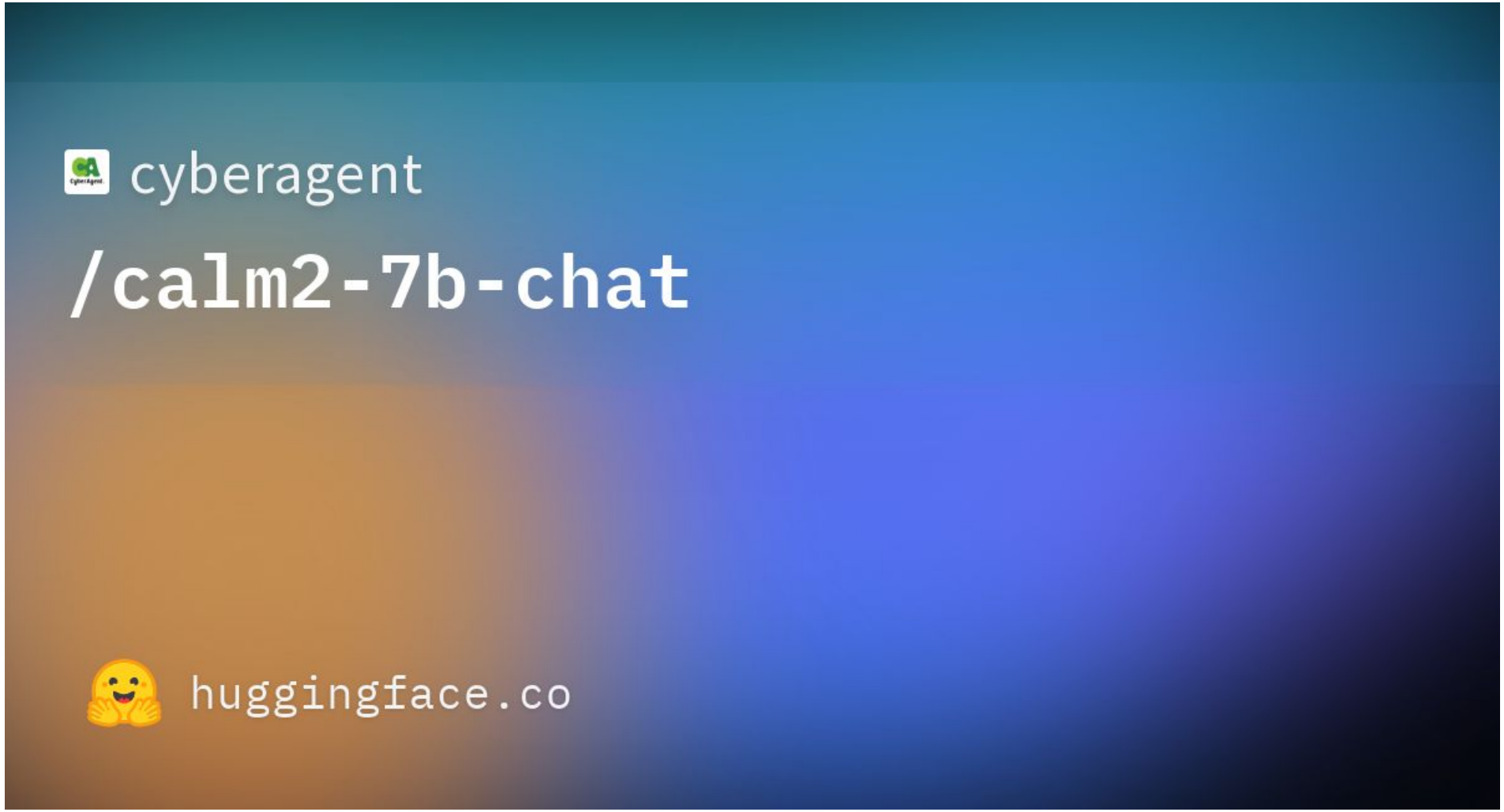
Kueue in CyberAgent

Workloads and Frameworks

- Training Machine Learning Models
  - batch/v1 Job
  - Kubeflow TFJob / PyTorchJob / MPIJob
- Jupyter Notebook
  - Managed by in-house system
    - Managed by in-house kueue-job manager
- Serving Machine Learning Model
  - Kserve (formerly KFServing)
  - Managed by ResourceQuota

Kueue in CyberAgent

Training Machine Learning Models: What do we work on the platform?

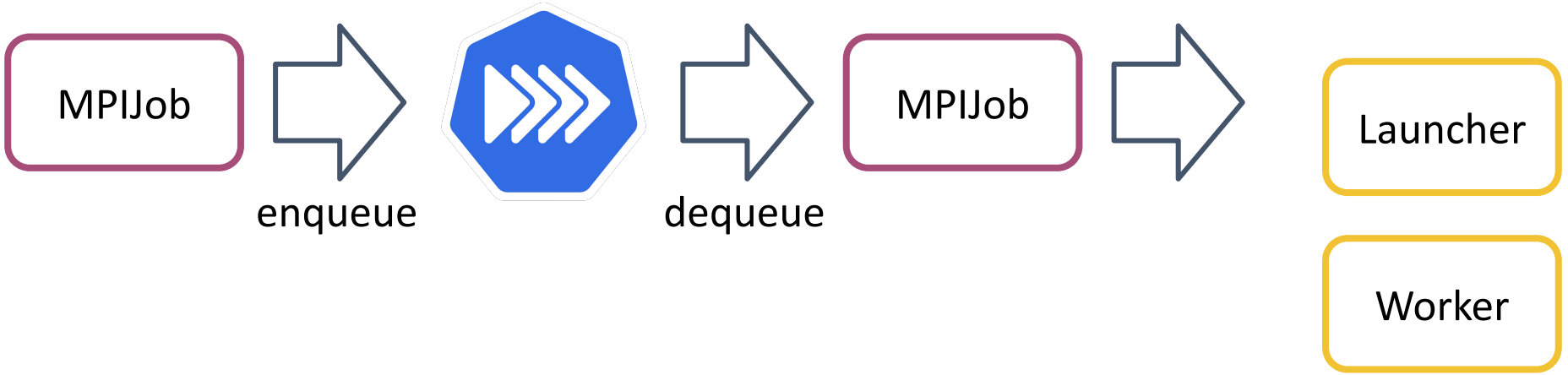


Kueue in CyberAgent

Training Machine Learning Model:

WaitForPodsReady with Configurable RequeueingStrategy

- When admitted Job isn't ready until timeout, Job is pushed back into ***Head*** (selected) or Tail of queue
- Not Ready Reasons:
  - Missing images, credentials and PVC etc.
  - The available quota is fragmented across multiple nodes

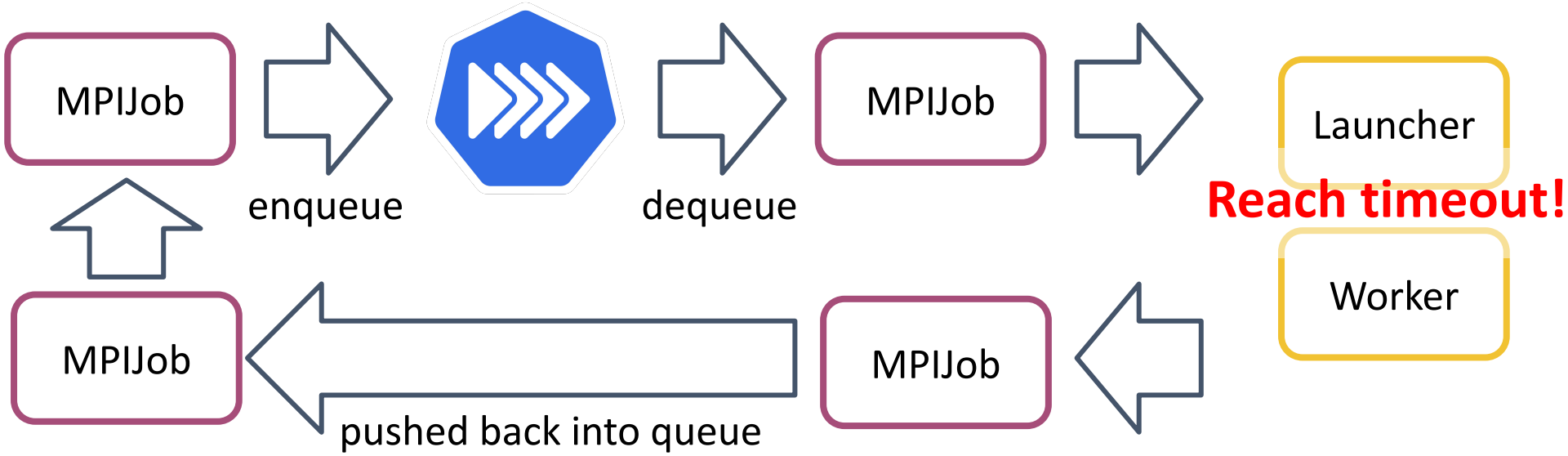


Kueue in CyberAgent

Training Machine Learning Model:

WaitForPodsReady with Configurable RequeueingStrategy

- When admitted Job isn't ready until timeout, Job is pushed back into ***Head*** (selected) or Tail of queue
- Not Ready Reasons:
  - Missing images, credentials and PVC etc.
  - The available quota is fragmented across multiple nodes



Kueue in CyberAgent

Bridging the Gap between Ideal and Real worlds

- Ideal world
  - All resources and workloads are completely managed by Kueue
- Real world
  - Conflicting between existing quota management system and Kueue
  - Some workloads and frameworks are not yet supported by Kueue
  - Kueue's functions are insufficient
    (e.g. Model Serving Server and Auto Scaling Semantics etc.)

Kueue in CyberAgent

Migrate K8s's ResourceQuota to Kueue's ClusterQueue

- Ideal world: Quota management and Queueing system by Kueue
  - Kueue admits jobs based on quota defined in ClusterQueues
- Real world: Existing quota management system
  - It depends on ResourceQuota
  - It distributes ResourceQuota to namespaces (tenants) to reserve quota

Kueue in CyberAgent

Migrate K8s's ResourceQuota to Kueue's ClusterQueue

- Ideal world: Quota management and Queueing system by Kueue
  - Kueue admits jobs based on quota defined in ClusterQueues
- Real world: Existing quota management system
  - It depends on ResourceQuota
  - It distributes ResourceQuota to namespaces (tenants) to reserve quota
- Gap
  - Over-assignment could occur
    - Over-assigned Job continues to be rejected by ResourceQuota until resources are free
  - Kueue can not admit jobs based on ResourceQuota

Kueue in CyberAgent

Migrate K8s's ResourceQuota to Kueue's ClusterQueue

- Ide
  -
- Re
  -
  -
  - It distribute[illegible] to namespaces (tenants) to reserve quota
- Gap
  - Over-assignment could occur
    - Over-assigned Job continues to be rejected by ResourceQuota until resources are free
  - Kueue can not admit jobs based on ResourceQuota

Approaches to bridge the gap:

1. WaitForPodsReady with Configurable RequeuingStrategy
2. AdmissionCheck for ResourceQuota

Kueue in CyberAgent

Migrate K8s's ResourceQuota to Kueue's ClusterQueue

- Ide
  -
- Re
  -
  -
  - It distribute… …to namespaces (tenants) to reserve quota
- Gap
  - Over-assignment could occur
    - Over-assigned Job continues to be rejected by ResourceQuota until resources are free
  - Kueue can not admit jobs based on ResourceQuota

Approaches to bridge the gap:

1. WaitForPodsReady with Configurable RequeuingStrategy
2. ***AdmissionCheck for ResourceQuota (※1)***

※1 Currently, it is evaluating in NON-production environment

Kueue in CyberAgent

Migrate K8s's ResourceQuota to Kueue's ClusterQueue

- WaitForPodsReady with Configurable RequeuingStrategy
  - Pros:
    - Ready to use by a Kueue's configuration setting
  - Cons:
    - Increased API-server load due to repeated try to create Pods and re-queueing jobs

Kueue in CyberAgent

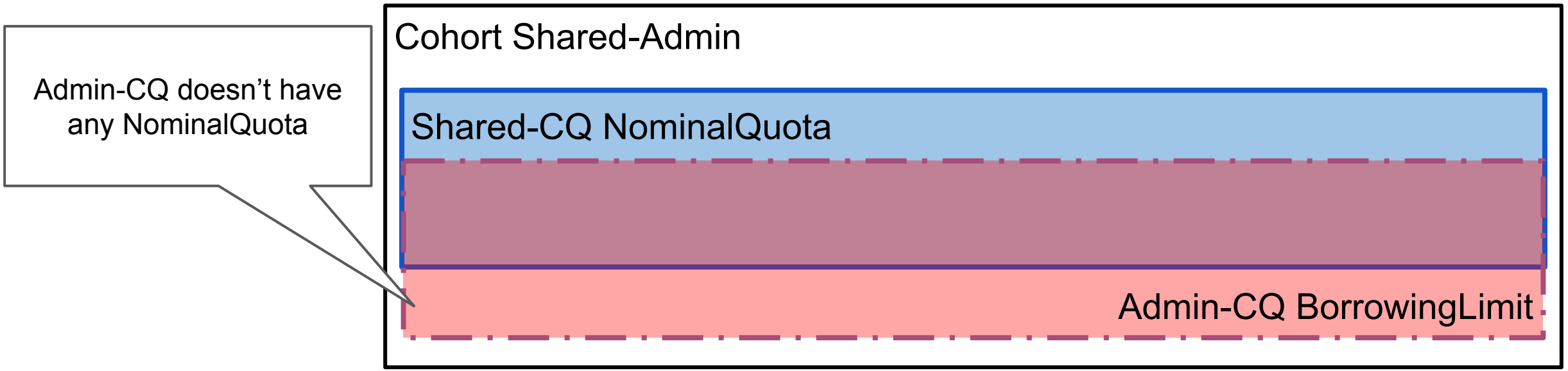
Migrate K8s's ResourceQuota to Kueue's ClusterQueue

- WaitForPodsReady with Configurable RequeuingStrategy
  - Pros:
    - Ready to use by a Kueue's configuration setting
  - Cons:
    - Increased API-server load due to repeated try to create Pods and re-queueing jobs
- AdmissionCheck for ResourceQuota
  - Pros:
    - It could avoid higher kube-api server load
  - Cons:
    - It needs to implement small custom controller

Kueue in CyberAgent

Overlapping Quota Model between User and Admin

- Conflicting Demands against GPUs
  - All GPUs always SHOULD be allocated to user's workloads
  - Cluster Admins WANT to verify platform features using GPUs
- Admins can submit Jobs only when there are some free quota in Shared-CQ

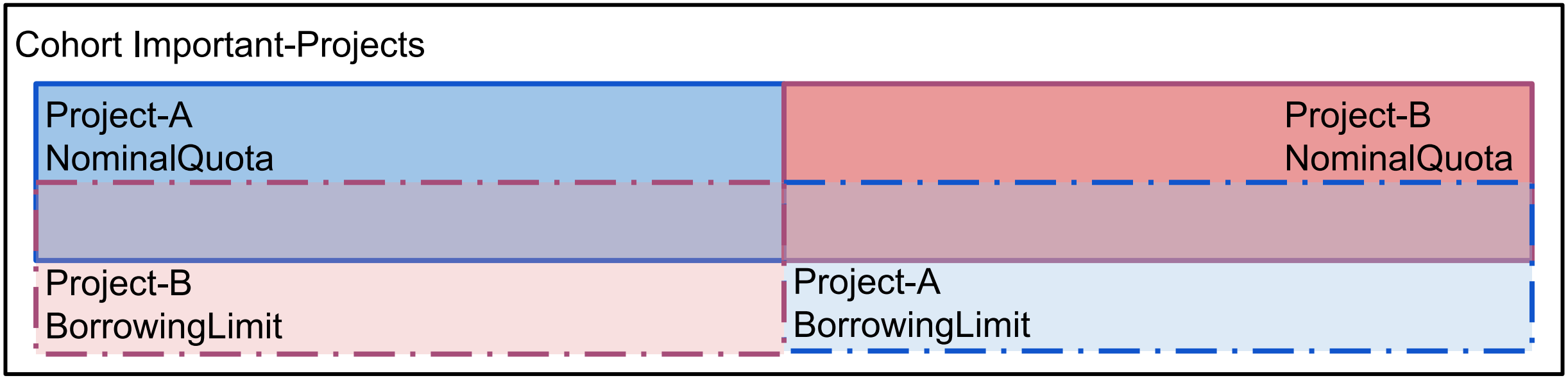


Kueue in CyberAgent

Elastically Reserved Quota Model between the same priority important projects

- Conflicting Demands against GPUs
  - Important projects WANT to reserve GPUs so that they can use GPUs when they want to use it
  - Any GPUs SHOULD NOT be left over for efficient usage
- Any Jobs borrowed by other ClusterQueues always be preempted

```
preemption:
    reclaimWithinCohort: Any
```



New features

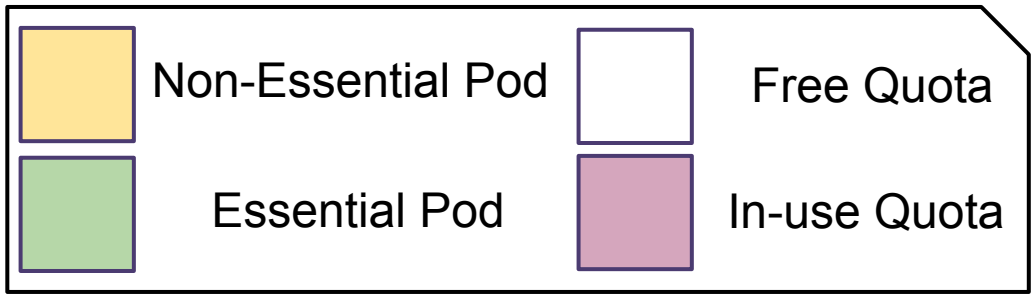
source

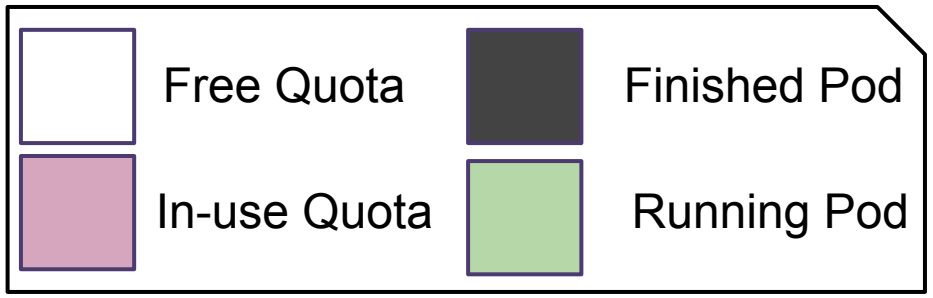
LendingLimit

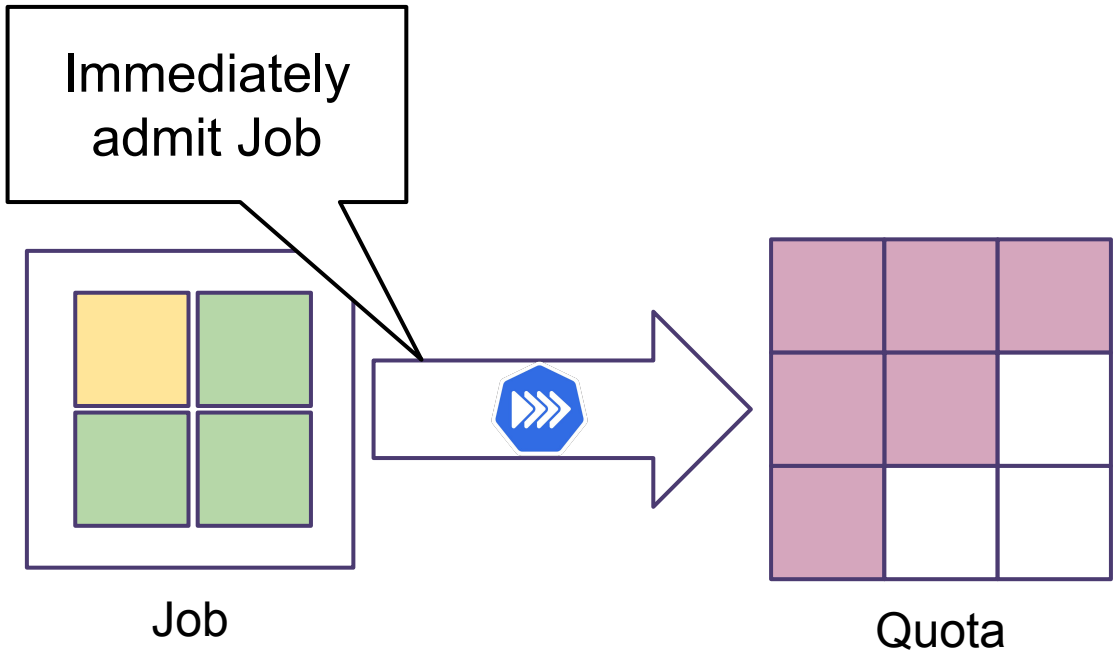
- **LendingLimit** indicates *how much resources can this ClusterQueue lend to other ClusterQueues within the same Cohort.*

- Use cases
  - Control the size of the guaranteed quota for a ClusterQueue in a Cohort
  - Reserve quota for latency sensitive workloads such as ML Model Serving
    - In the future, it is possible to integrate with Kserve (formerly KFServing)

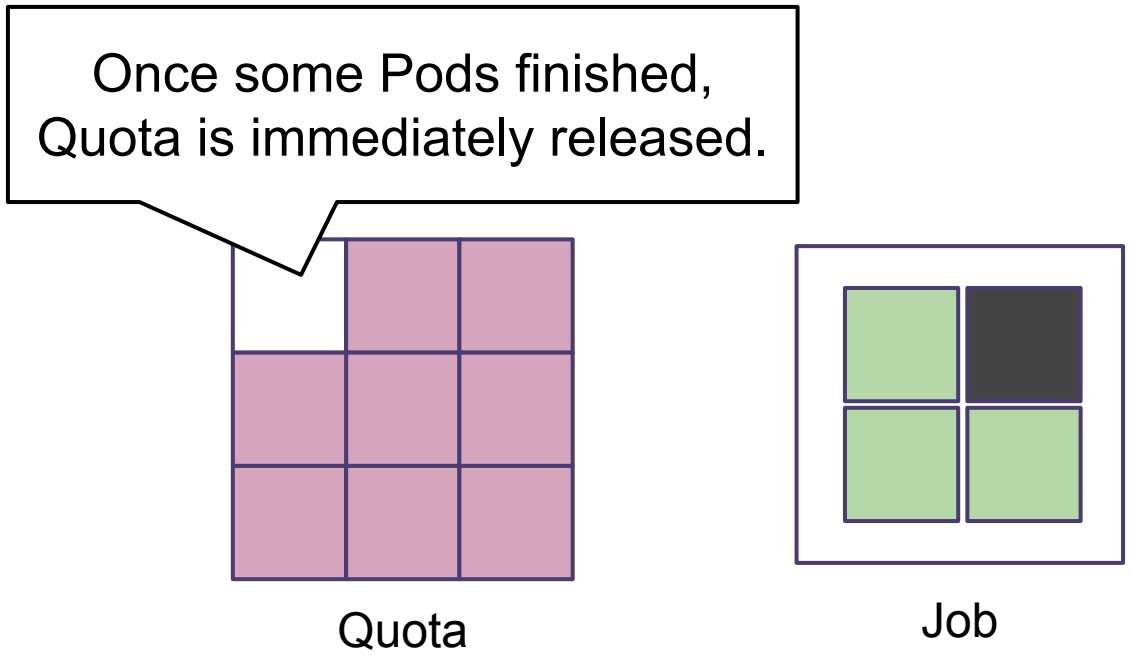
Dynamically Admission and Reclaiming







Partial Admission



Dynamically Reclaiming Resources

Provisioning Request integration

Motivation:

- All-or-nothing semantics
  - Currently Cluster-Autoscaler does not create new nodes until pods are created
  - Scale ups for large jobs may fail due to GPU stockouts and having 99% of nodes is not enough

Provisioning Request integration

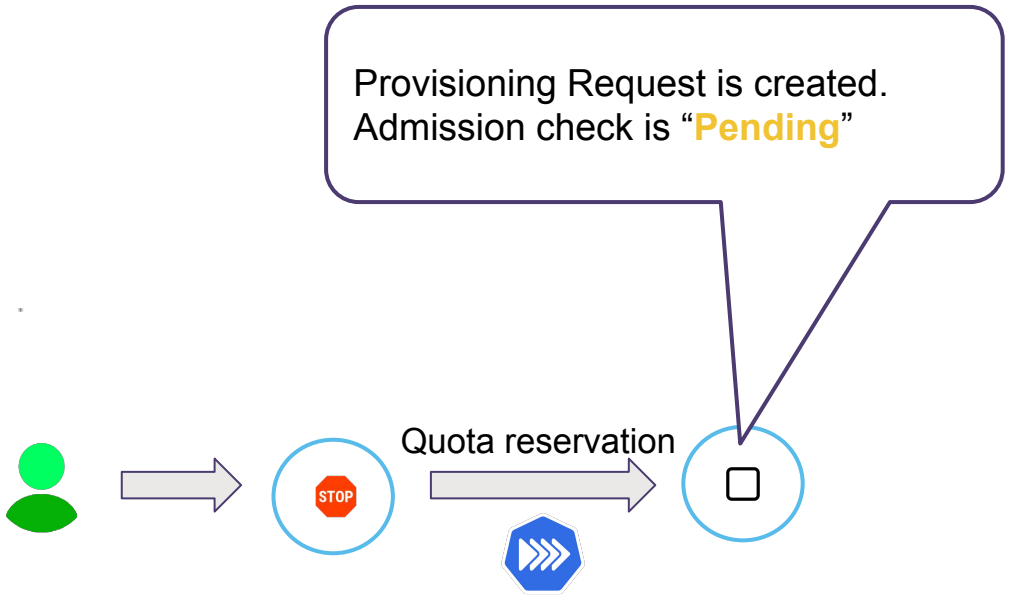
Motivation:

- All-or-nothing semantics
  - Currently Cluster-Autoscaler does not create new nodes until pods are created
  - Scale ups for large jobs may fail due to GPU stockouts and having 99% of nodes is not enough

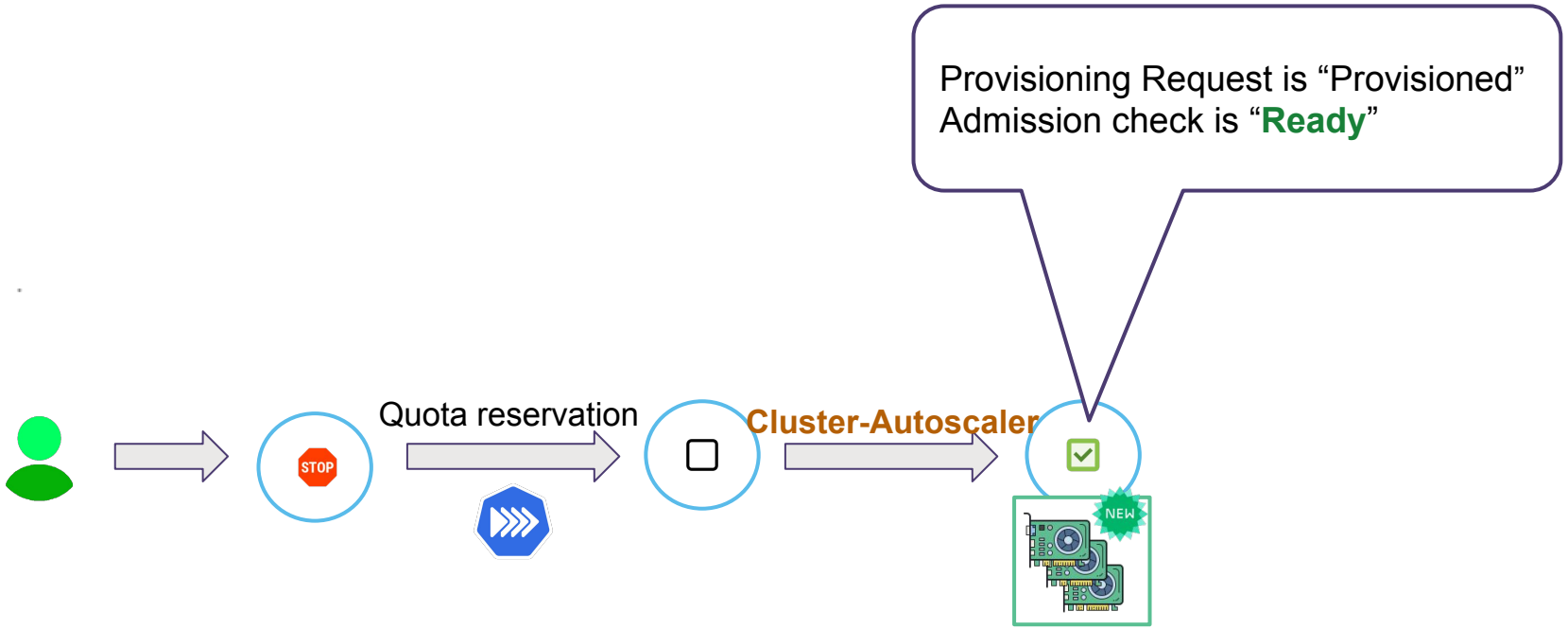
ProvisioningRequest Spec:

- PodSets - represent group of pods needing nodes
- ProvisioningClassName - describes the mode of provisioning
- Parameters - Parameters contains all other parameters classes may require

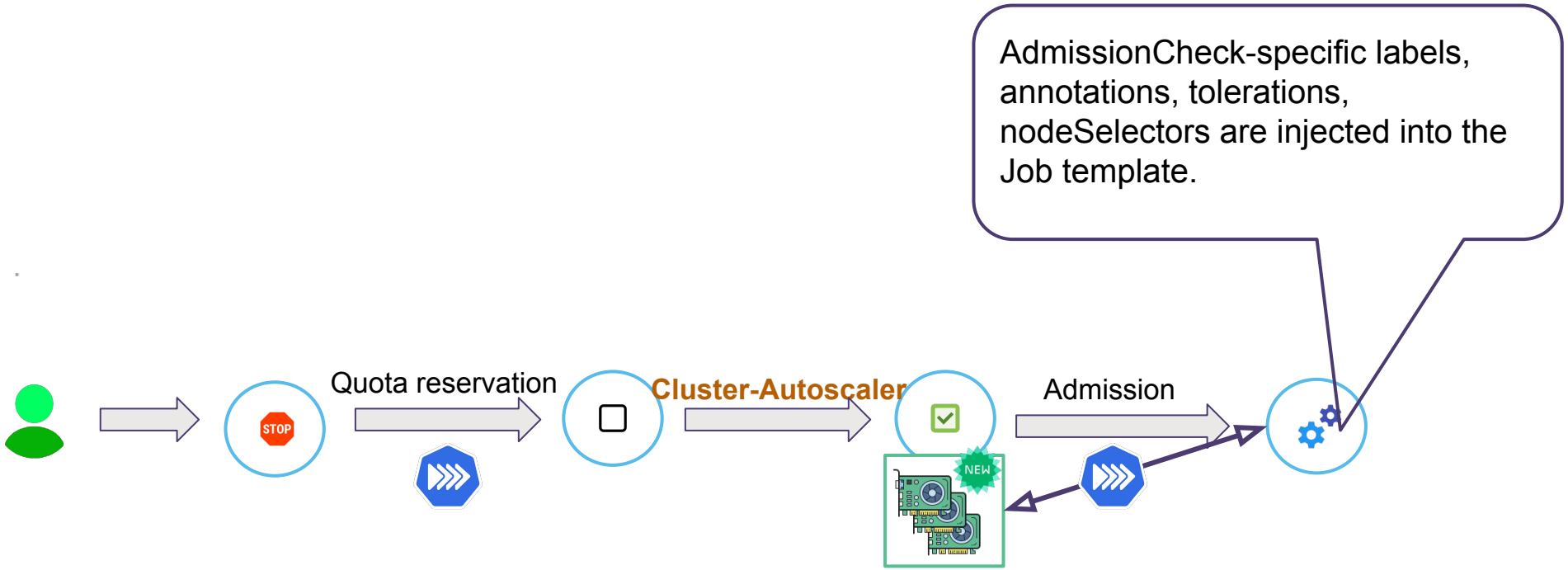
Provisioning Request integration



Provisioning Request integration



Provisioning Request integration

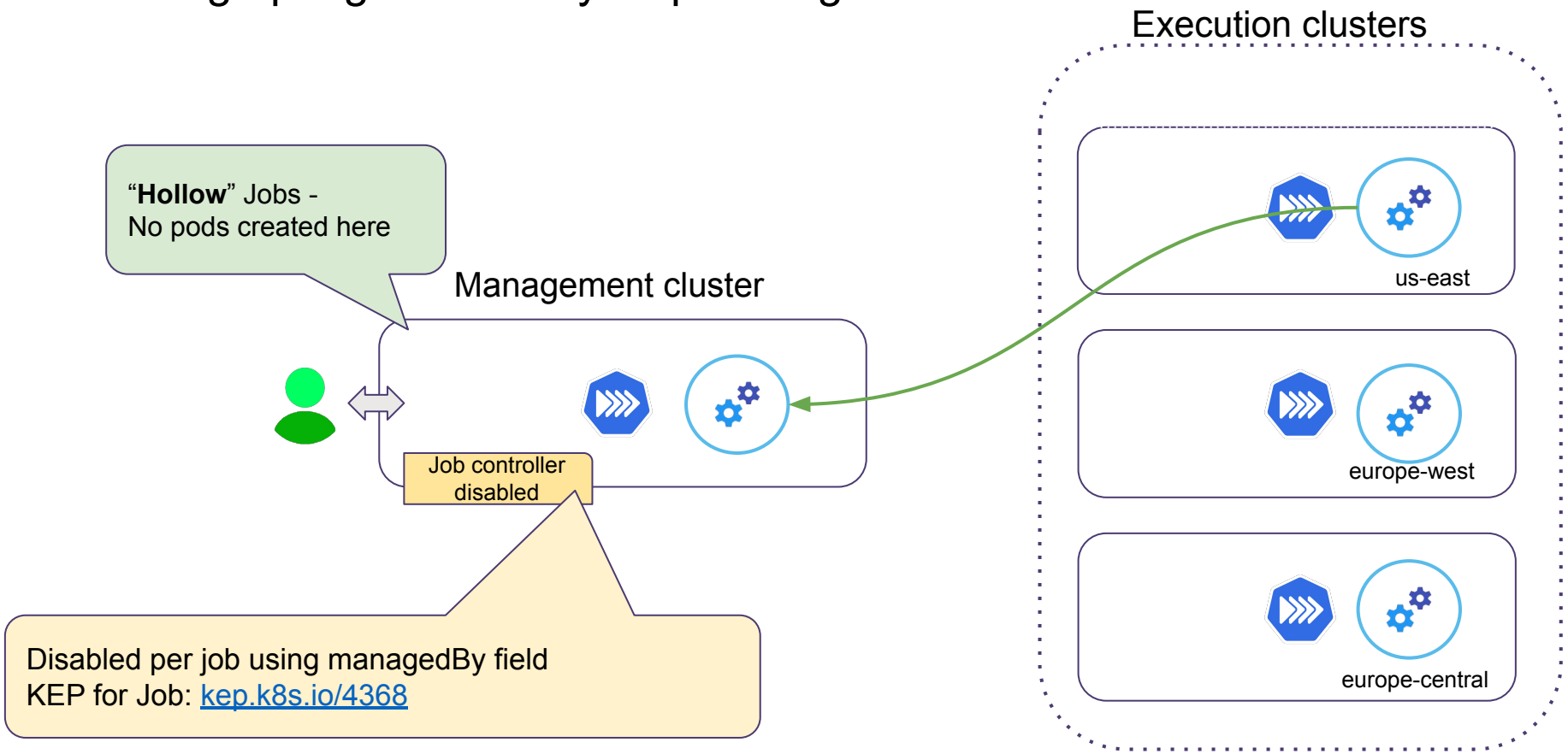


Multi-cluster Job dispatching

A.k.a.: **MultiKueue**

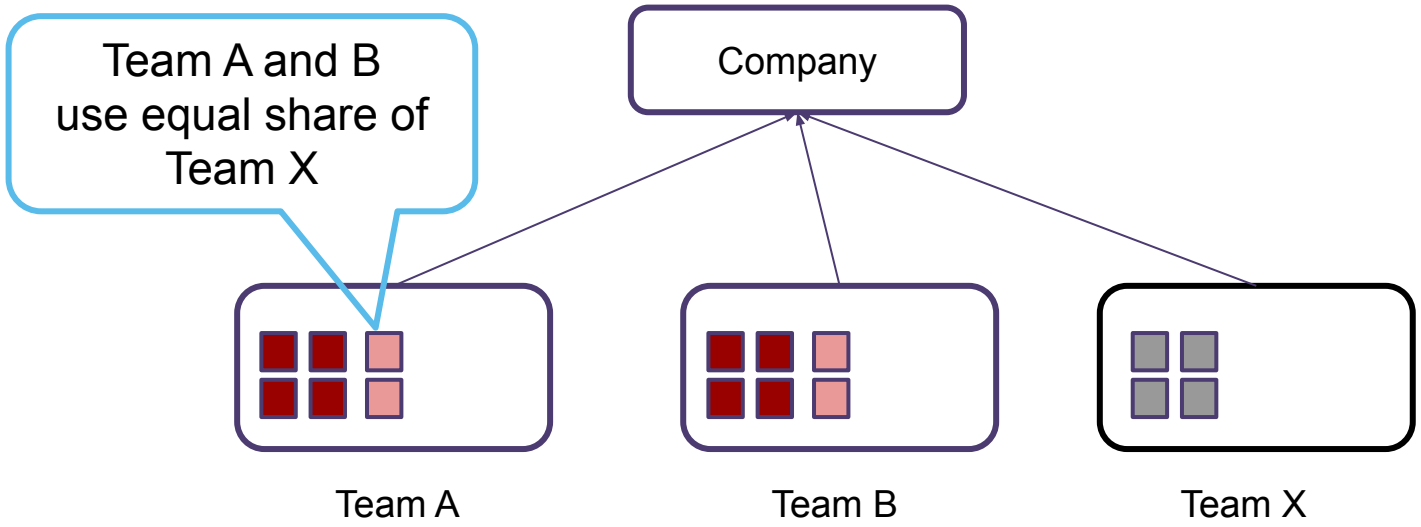
Motivation:

- Obtainability of GPUs across regions (or cloud providers)
- Scaling up big clusters by dispatching execution



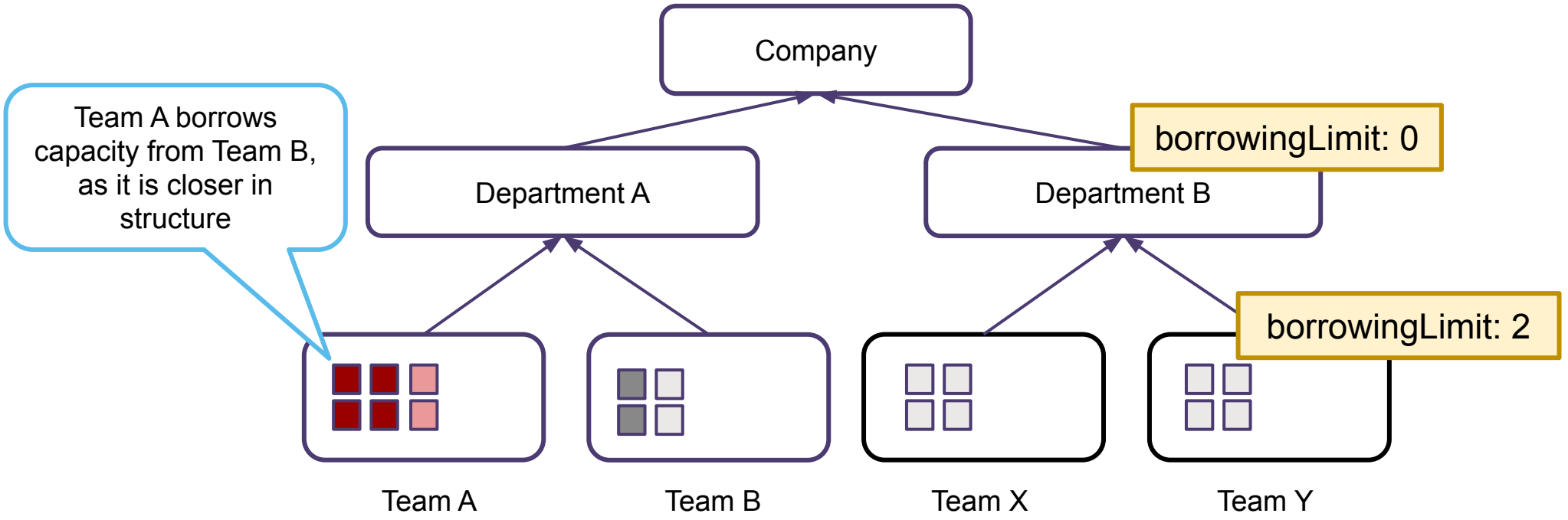
Fair sharing

- Gives fair access to unused resources to different teams
  - Without Fair sharing FIFO-scheme is used
- Resolve quota imbalances within structure by preemption



Hierarchical cohorts

- More levels of hierarchy to reflect the structure for large organizations
- Allows to specify `borrowingLimit` and `lendingLimit` at different levels
- Borrowing closely in the structure is prioritized



Try it yourself

Try yourself: https://kueue.sigs.k8s.io/ !

Getting involved

How to find us?

- slack.k8s.io #wg-batch
- 2 Biweekly meetings:
  - Thursdays 3pm CET
  - Thursdays 3pm PT
- git.k8s.io/community/wg-batch

WG-Batch Updates: What's New and What Is Next? - Yuki Iwai & Michał Woźniak

Questions

Michał Woźniak
- Email: michalwozniak@google.com
- GitHub: mimowo
- Slack: mimowo

Yuki Iwai
- Email: yuki.iwai.tz@gmail.com
- GitHub: tenzen-y
- Slack: tenzen-y

Please scan the QR Code above to leave feedback on this session