NEBIUS

One GPU Pool to Serve Them All: Scheduling with Complex Priorities

Yuriy Natarov
Team Lead, Nebius Token Factory

Dedicated inference endpoints

Guaranteed GPU capacity per customer, with the ability to scale beyond the quota during peak load

Public inference endpoints

Shared GPU reserve, with the ability to scale beyond the quota for bursts of customer requests

One GPU pool

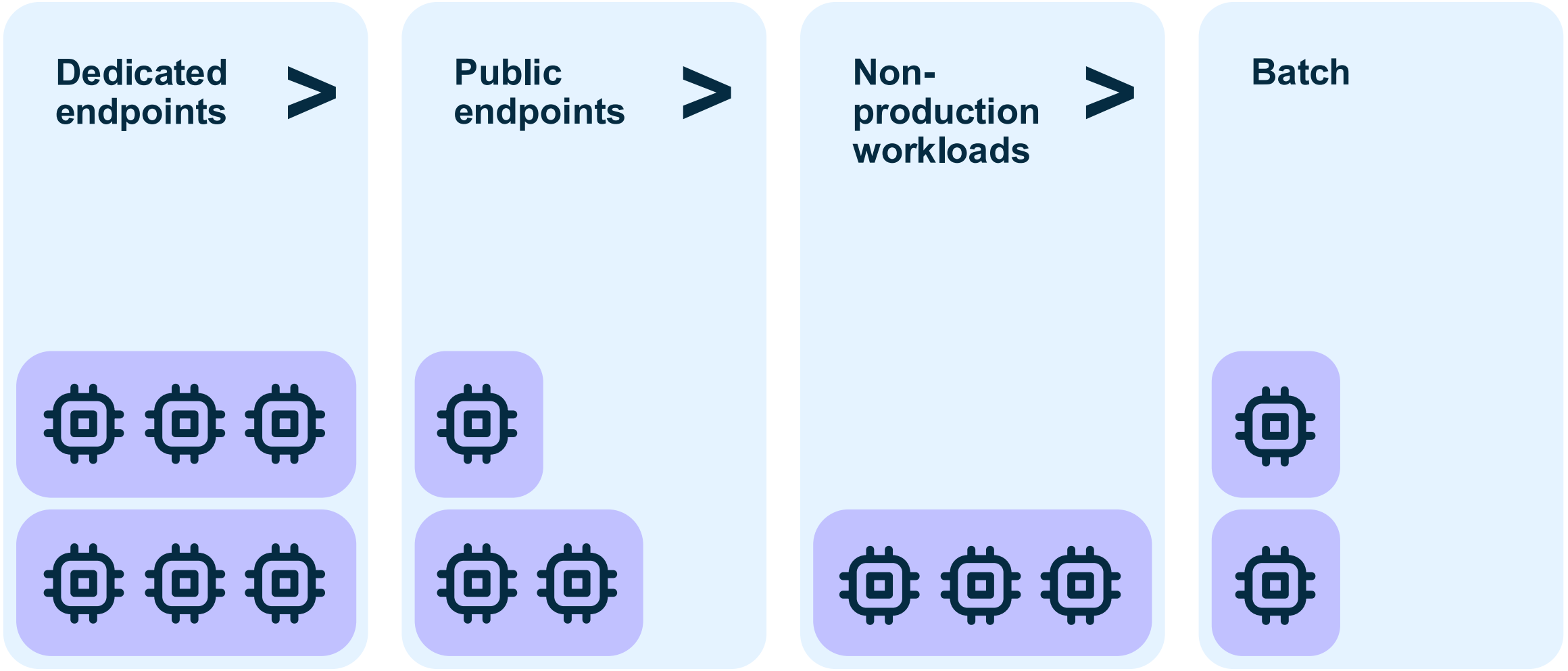
Batch inference

Processes a backlog of tasks, ideally during off-peak hours, with a 24-hour completion target

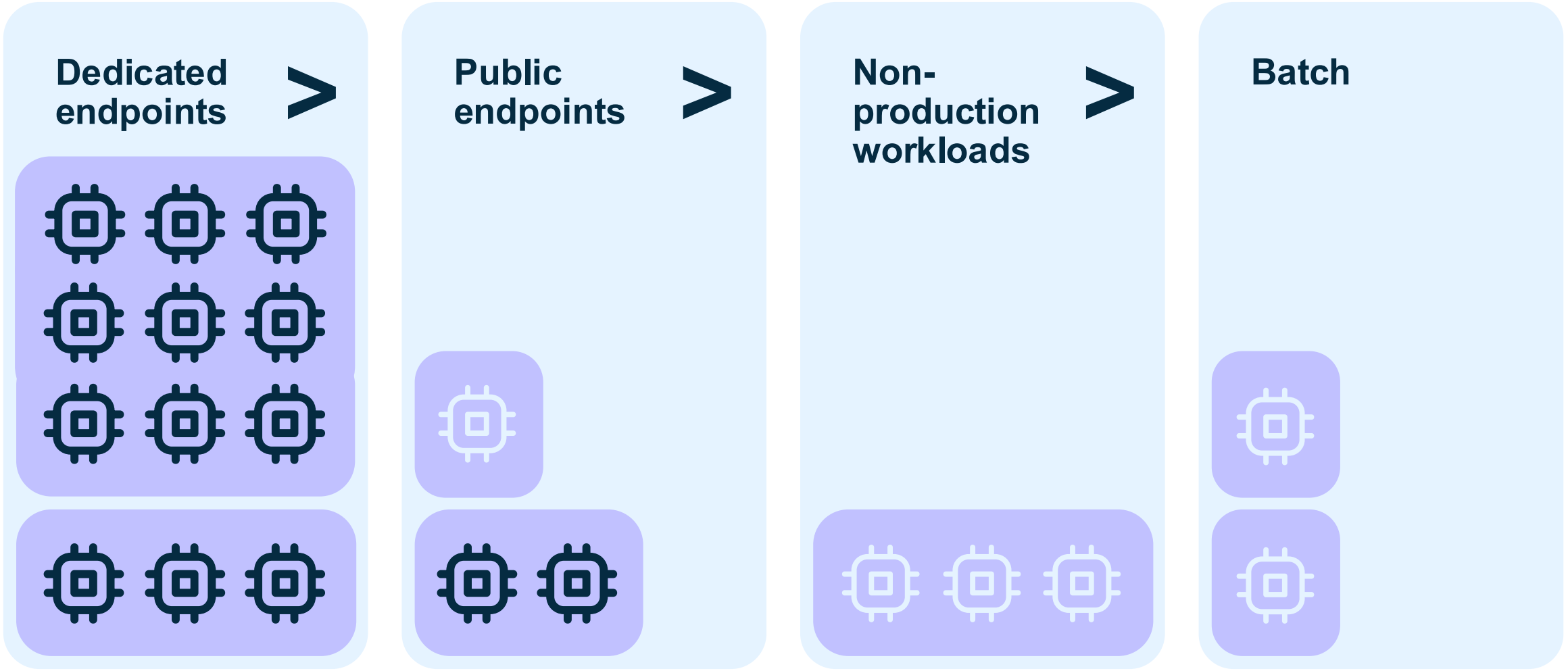
Non-production workloads

Occasional inference workloads for internal use

Can we just assign priorities?

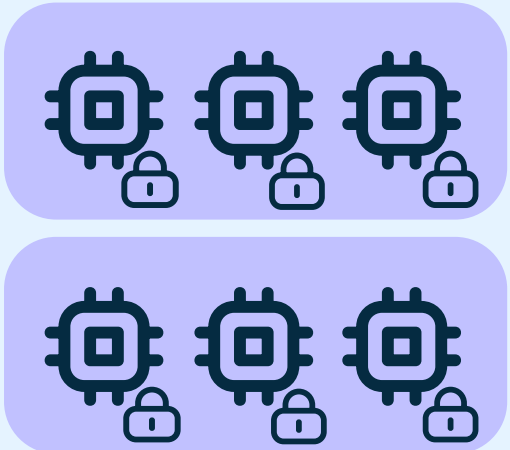


Can we just assign priorities? Nope!

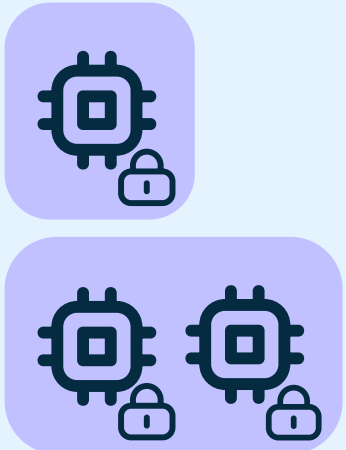


Let’s have quotas and prioritize the rest

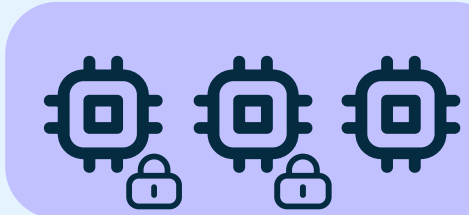
Dedicated endpoints >



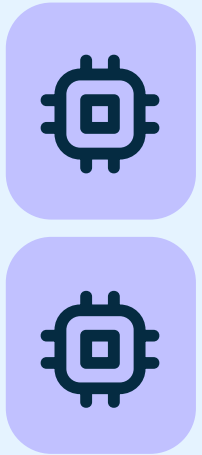
Public endpoints >



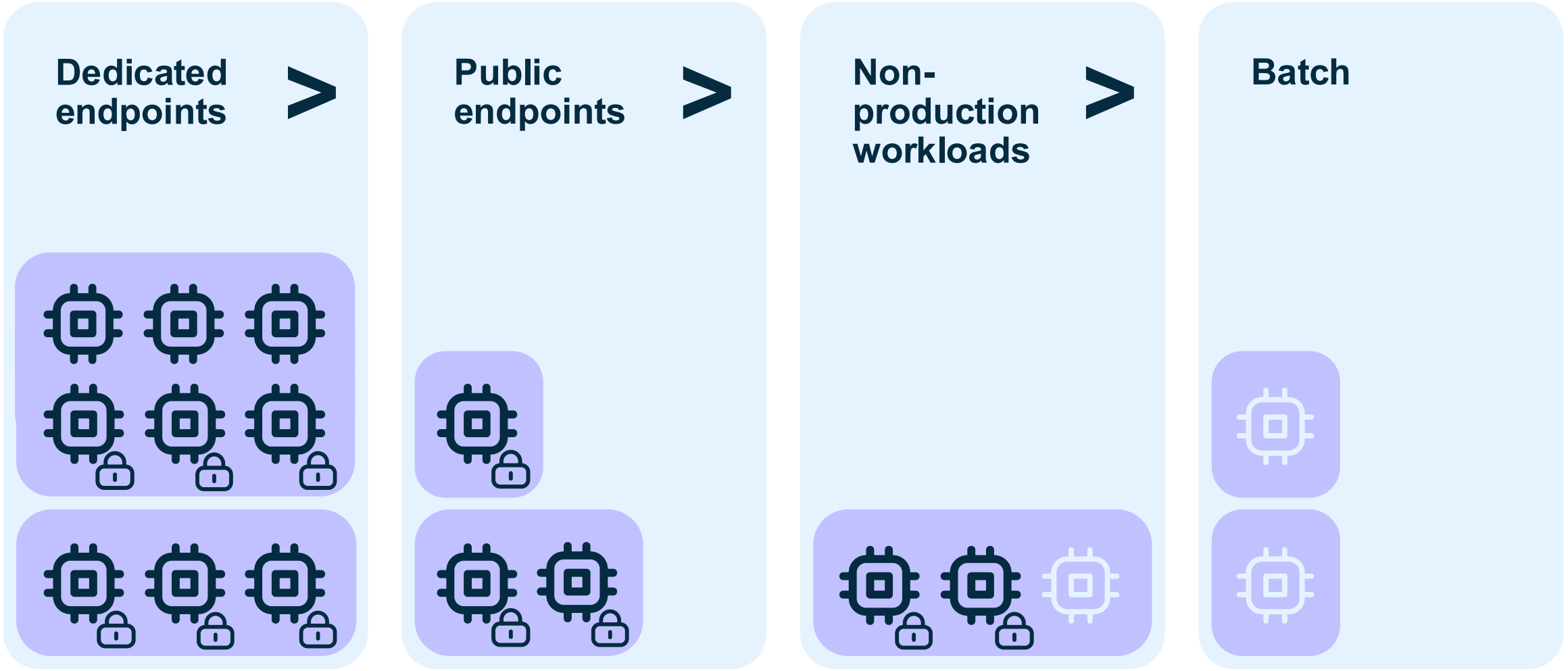
Non-production workloads >



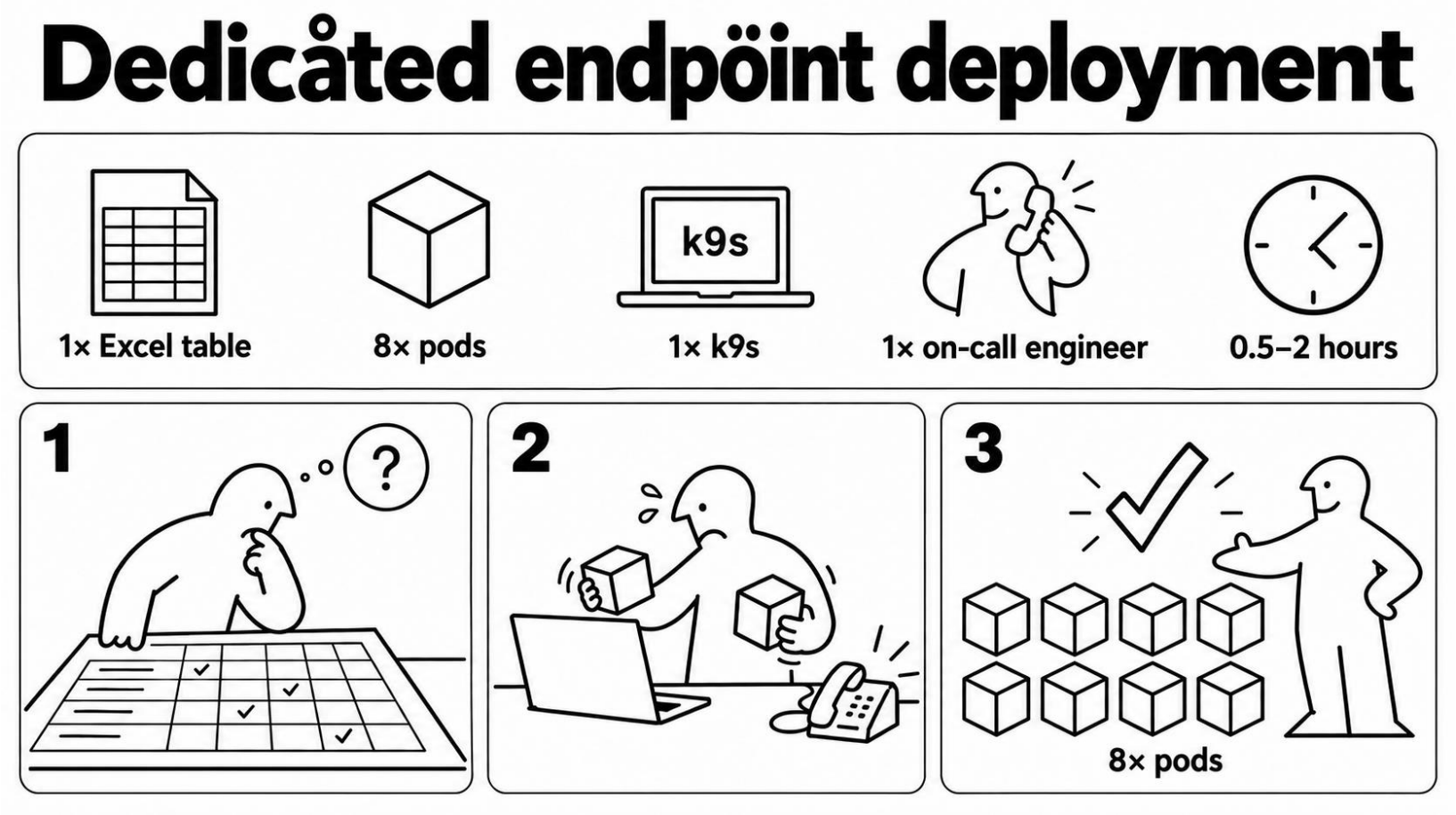
Batch



Let’s have quotas and prioritize the rest



Option 1: Manual :)



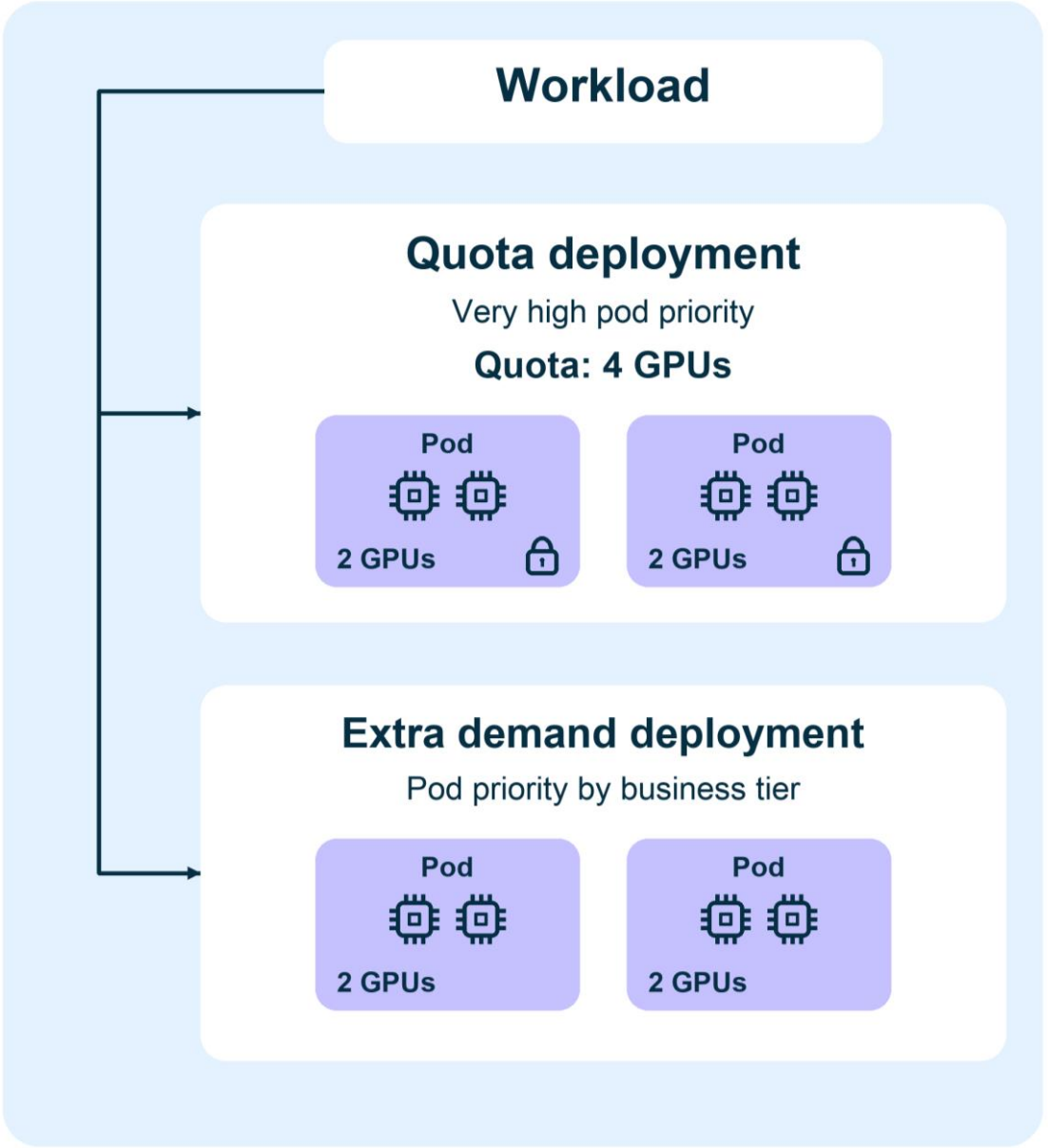
Option 2: Two deployments

Pros

1. Native Kubernetes priority and preemption

Cons

1. Rolling updates require coordination
2. Autoscaling must account for replicas in both deployments
3. Changing the quota may require replacing running pods
4. Splitting a single workload across two deployments limits architectural flexibility



Option 3: Advanced scheduling solution

Requirement	KAI	Kueue	Volcano	YuniKorn
Priority-based preemption above GPU quota	Native	Native	No	No

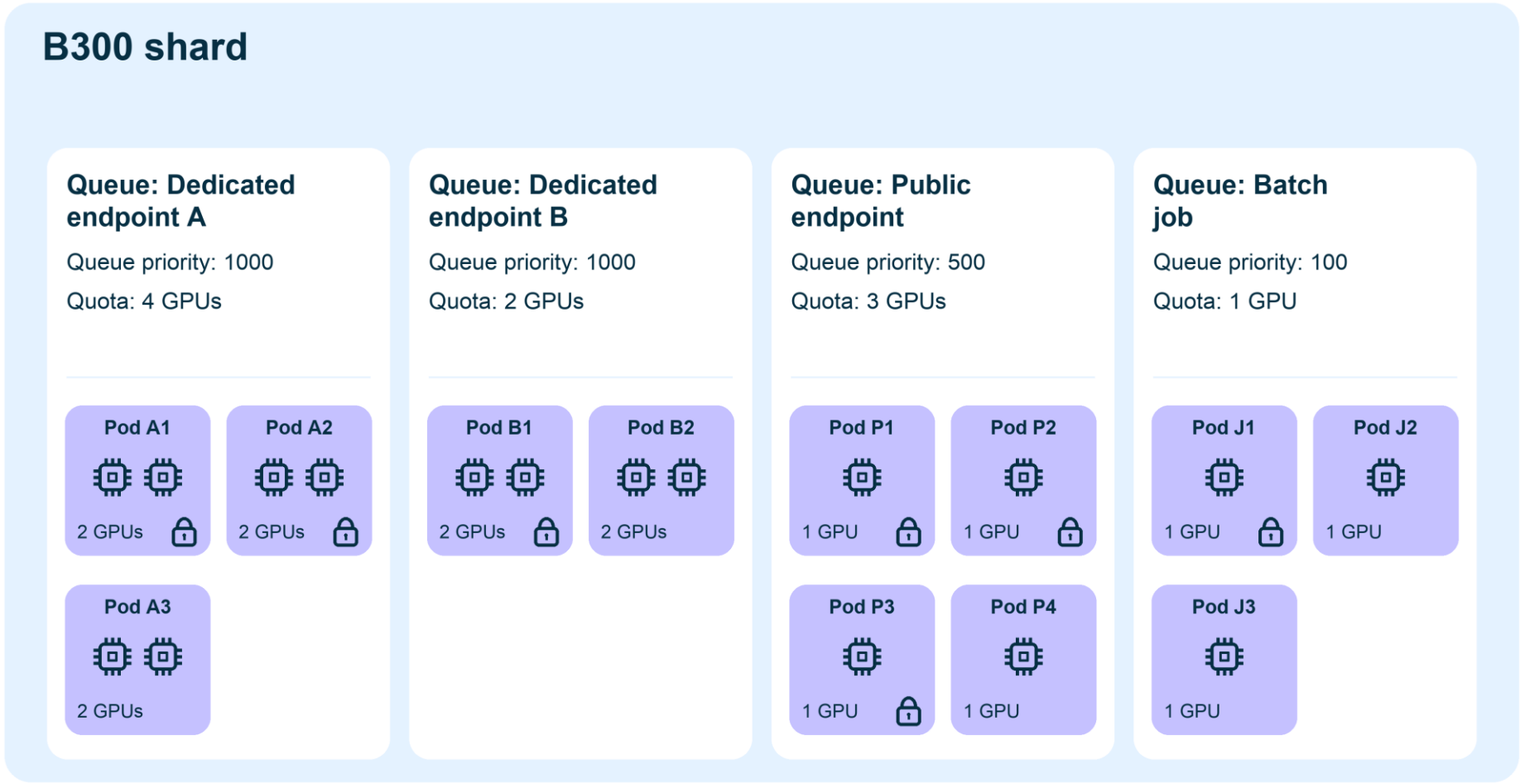
Option 3: Advanced scheduling solution

Requirement	KAI	Kueue	Volcano	YuniKorn
Priority-based preemption above GPU quota	Native	Native	No	No
NUMA-aware GPU and CPU scheduling	Native plugin	Requires an external NUMA-aware scheduler	CPU only	No
Configurable preemption delay	Native	No	No	Native
Grove integration	Supported	Partial	Partial	No

KAI: Shards, queues and quotas



KAI: Shards, queues and quotas



Results

1. Efficient utilization of spare GPU capacity
2. No manual capacity management: provisioning takes minutes
3. Better quality of service during request bursts
4. Guaranteed GPU capacity is reclaimed automatically



Thank you!

Contact me:

- linkedin.com/in/acuion
- acuion@nebius.com

Feedback

One GPU Pool to Serve Them All: Scheduling with Complex Priorities