KubeCon CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024


# Implementing Fine-Grained and Pluggable Container Resource Management Leveraging NRI

*Qiang Ren*
*Intel*

*He Cao*
*ByteDance*

# Agenda



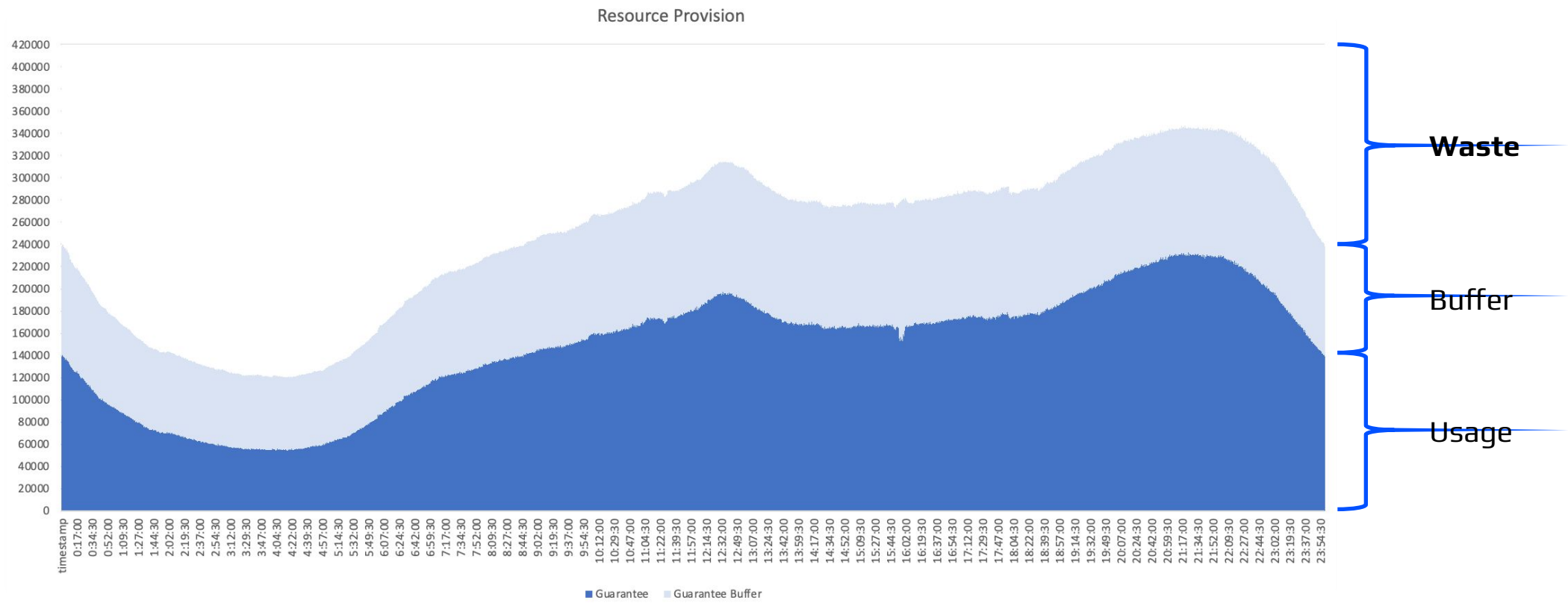
- Katalyst Overview
- Plugin-Based Resource Management Mechanism of Katalyst
- NRI Mechanism
- Application of NRI in Katalyst
- Community

# 1 Katalyst Overview

# Capacity Planning Challenges



- The resource utilization of online services exhibits a tidal pattern, with very low utilization during the night
- Users tend to over-request resources to ensure service stability, leading to resource wastage

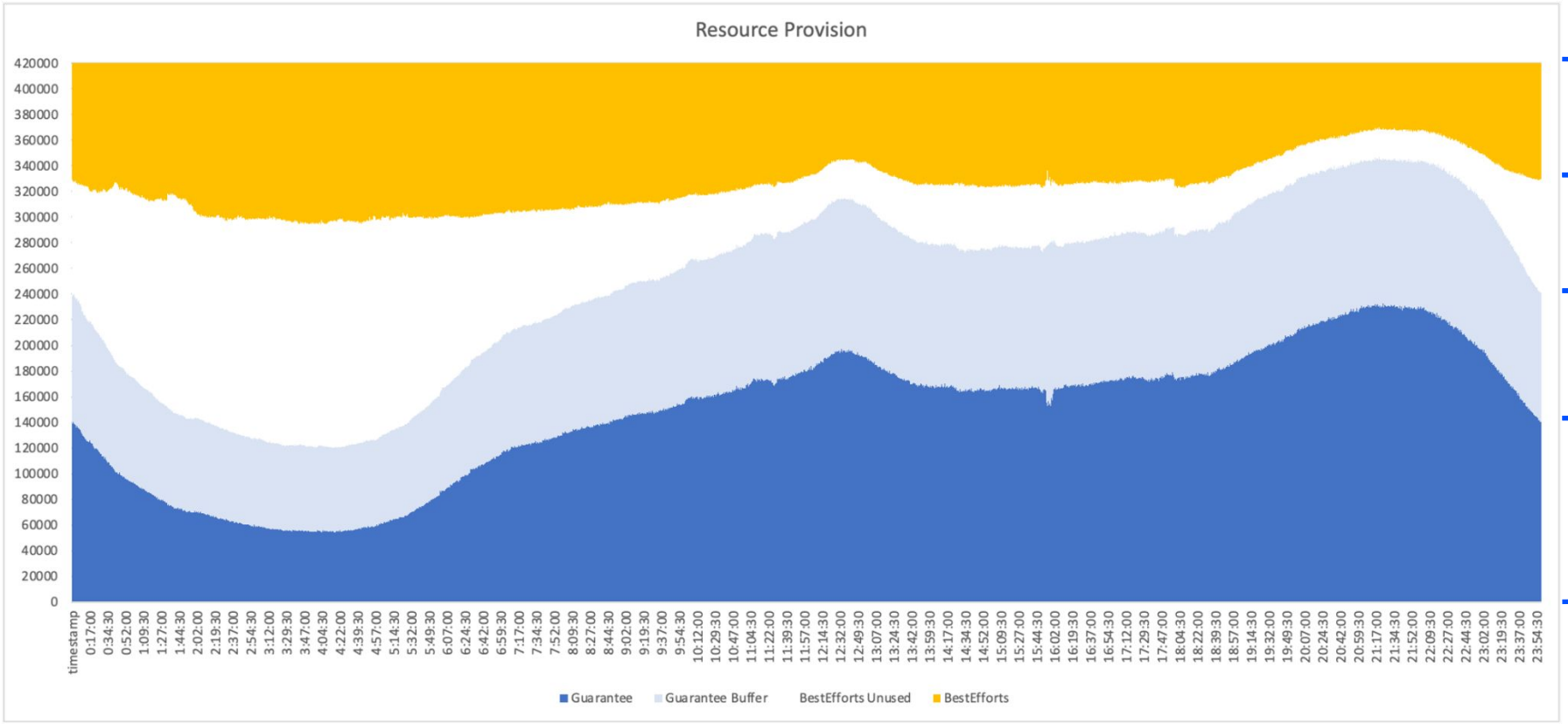


# Colocation



The resource utilization patterns of online services and batch jobs are inherently complementary:

- Online services prioritize CPU and RPC latency
- Batch jobs prioritize memory and throughput




Batch jobs' usage

Not used by batch jobs

**Reclaimed resources**

Buffer

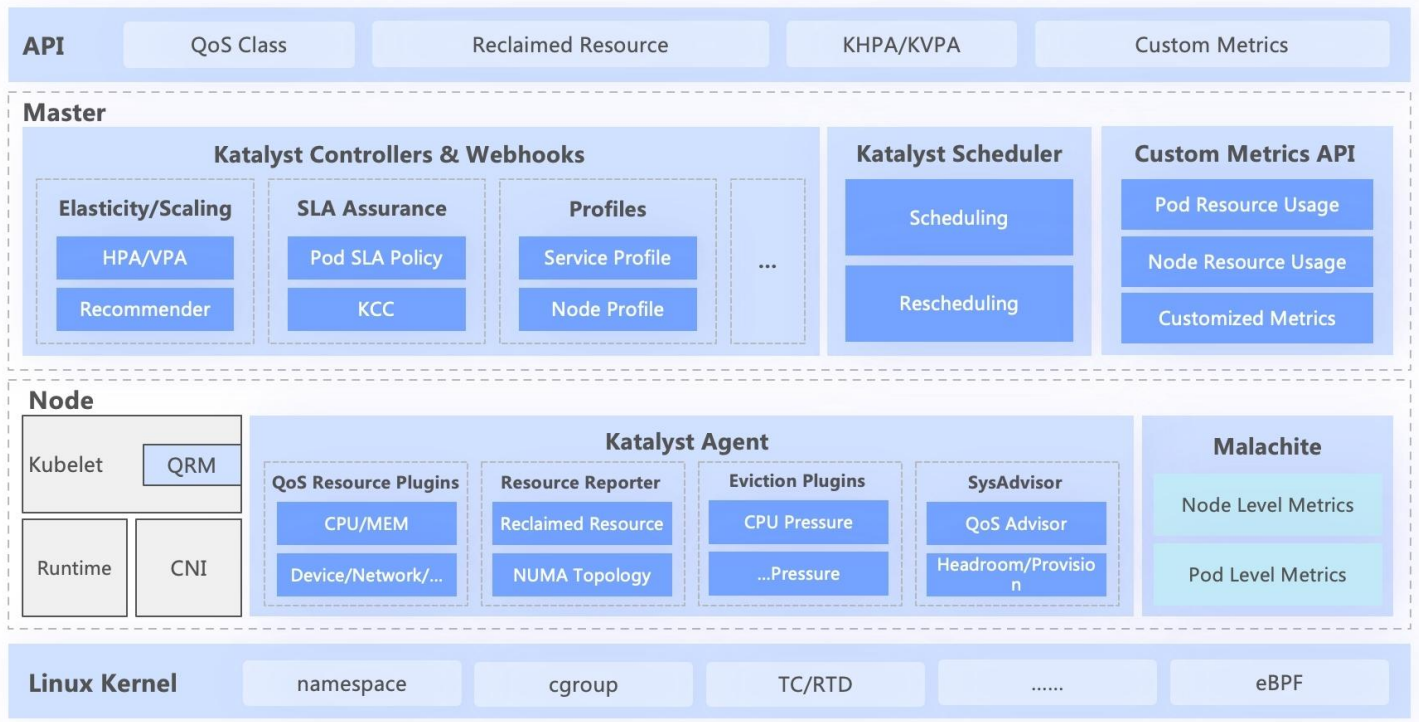
Online services' usage

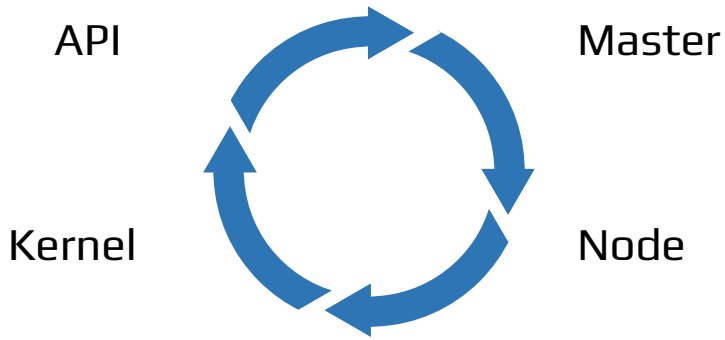
# Katalyst: Resource Management System



Katalyst

Katalyst, derived from the “catalyst” in chemical reactions, provides enhanced resource management capabilities for workloads running on Kubernetes




https://github.com/kubewharf/katalyst-core

# 2 Plugin-Based Resource Management Mechanism of Katalyst

# Fine-Grained Resource Management Strategies



## 4 Extended QoS Classes

- Expressing services' requirements for resource quality
- Naming based on CPU as the primary resource dimension

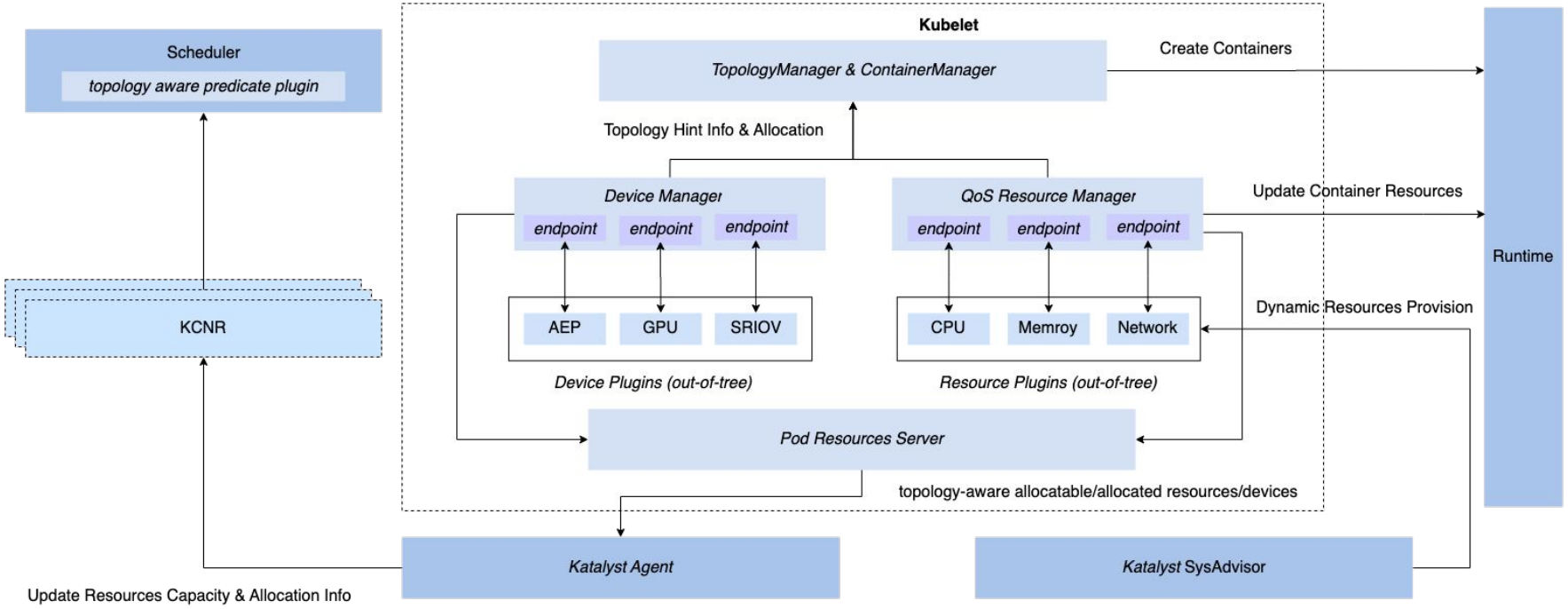
## More QoS Enhancements

- NUMA binding
- NUMA exclusive
- Network class
- ...

| QoS Classes | Attributes | Suitable for workload types | Relationship with K8s QoS |
|---|---|---|---|
| **dedicated_cores** | • Dedicated CPU cores, not shared with other workloads<br>• Supports binding to NUMA nodes for improved performance | Extremely latency-sensitive workloads, such as ads, search, and recommendation | Guaranteed |
| **shared_cores** | • Shared CPU pool<br>• Supports further dividing CPU pools based on business types<br>• Also supports NUMA binding | Workloads that can tolerate a certain degree of CPU throttling or interference, such as microservices | Guaranteed/ Burstable |
| **reclaimed_cores** | • Over-committed resources<br>• Resource quality is relatively unguaranteed<br>• May be evicted | Workloads that are not sensitive to latency and prioritize throughput, such as model training and batch jobs | BestEffort |
| **system_cores** | • Reserved CPU cores<br>• Ensure the stability of system components | Critical system agents | Burstable |

# QRM Framework: Making Kubelet's Resource Management Strategies Extensible






## QoS Resource Manager

- Enabling plugin-based resource management capabilities
- Integrating with Topology Manager to achieve NUMA affinity
- Dynamically adjust resources while containers are running

## Resource Plugin

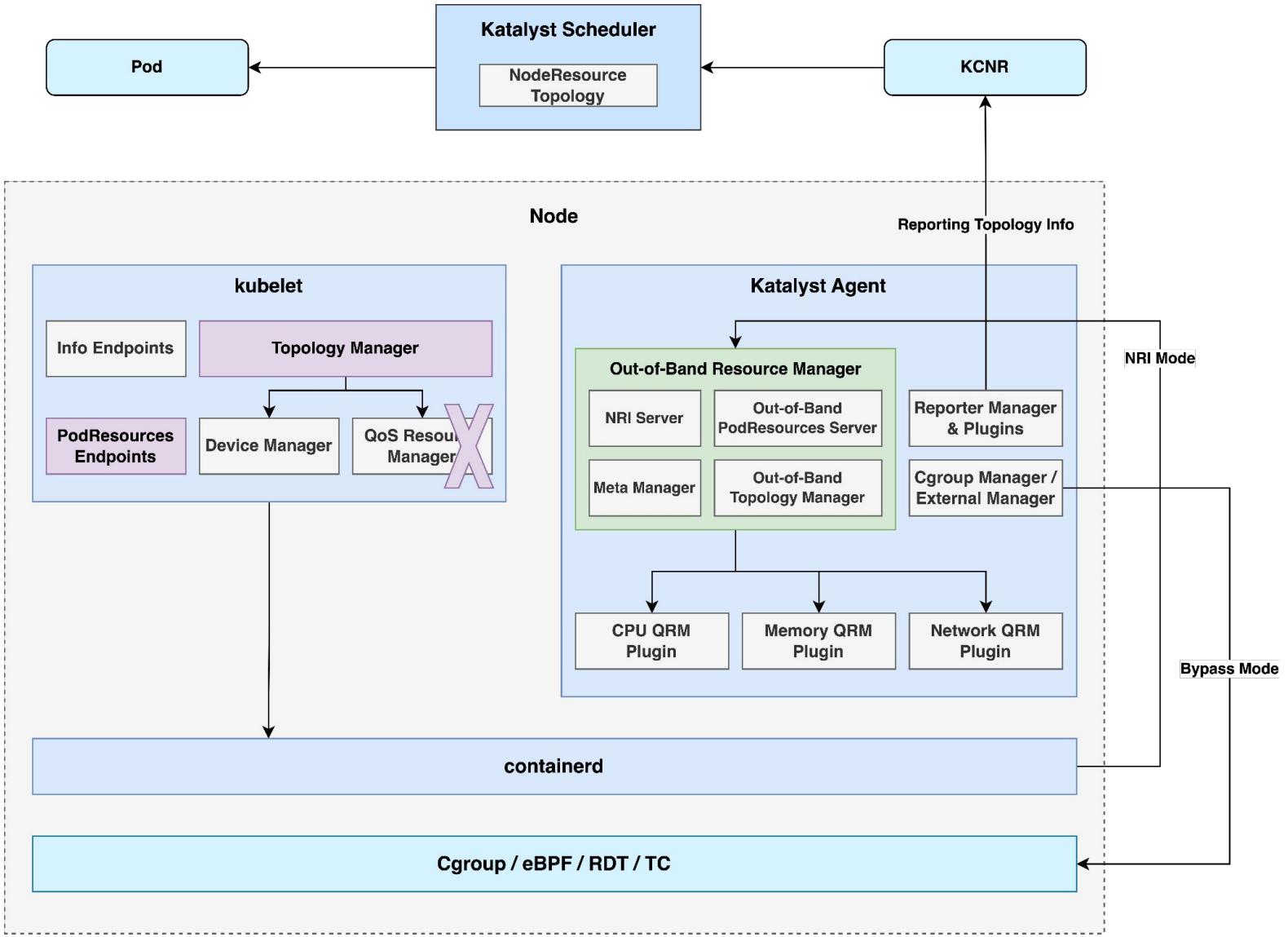
- Customizing resource allocation policies for containers through plugins
- Collecting and reporting topology information

# ORM Framework: Decoupling the QRM Framework from Kubelet



## ORM: Out-of-Band Resource Manager

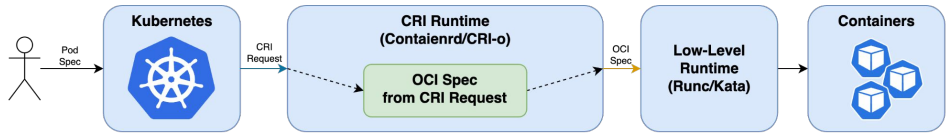
- **2 Modes**
  - NRI mode
  - Bypass mode
- **Supporting NUMA Affinity**
  - Out-of-band Topology Manager
  - Out-of-band PodResources Server



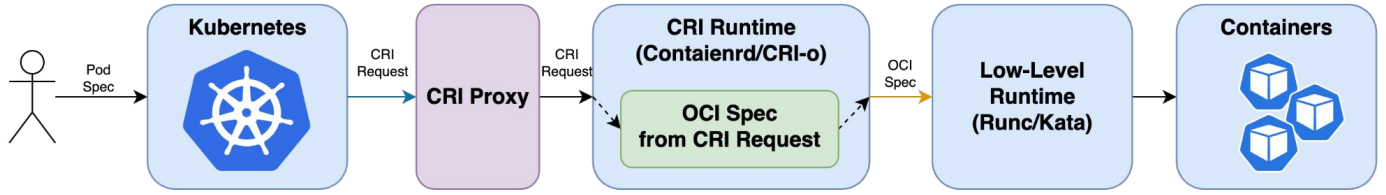
# 3 NRI Mechanism

# Common Methods for Kubernetes QoS

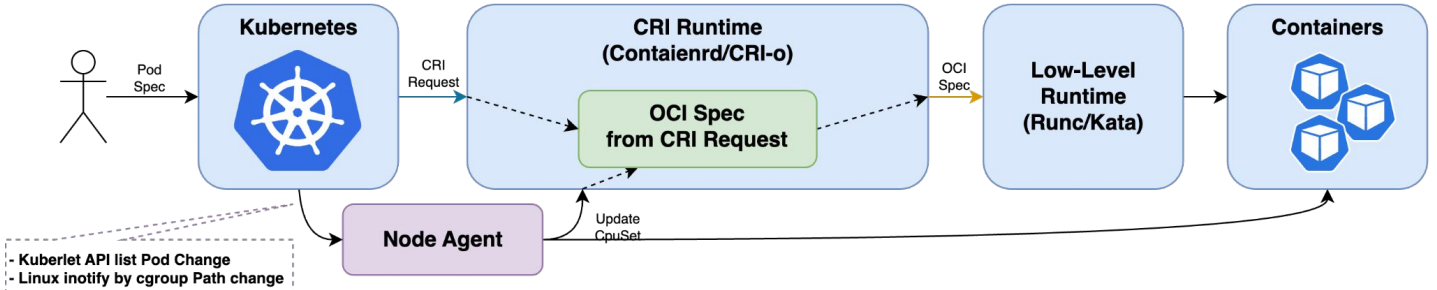




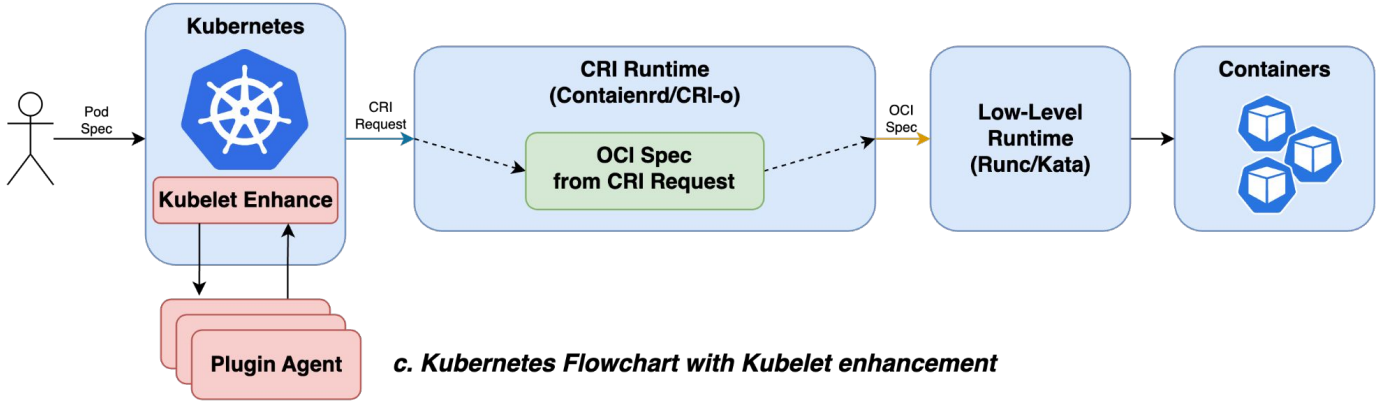

*Kubernetes Default Flowchart*




*a. Kubernetes Flowchart with CRI Proxy*




*b. Kubernetes Flowchart with Standalone Agent*

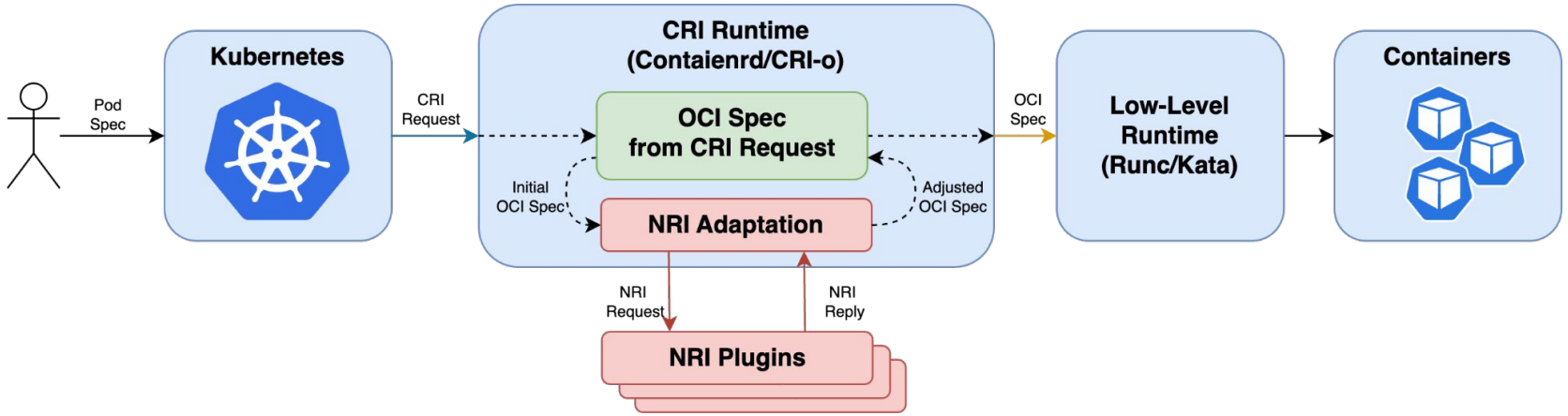



*c. Kubernetes Flowchart with Kubelet enhancement*

# NRI: A Powerful Tool for Fine-Grained Resource Management on Nodes



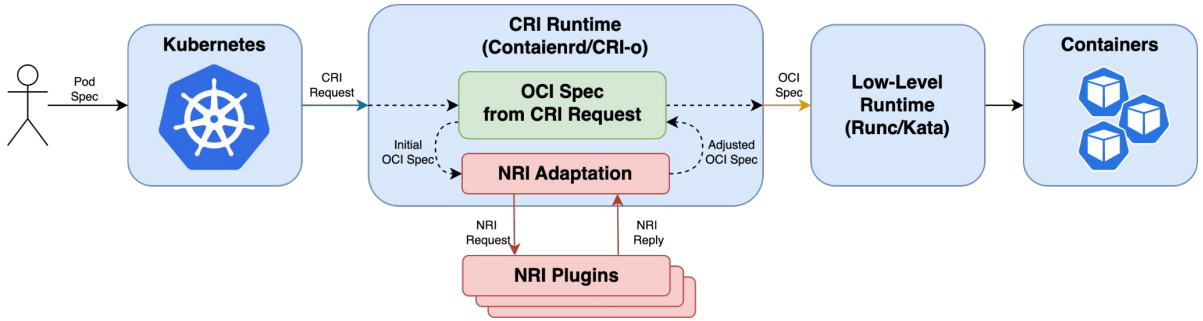
- NRI is a common framework for managing CRI runtime plugins, applicable to container runtimes such as containerd and CRI-O.
- NRI provides extension plugins with a basic mechanism to track container states and make limited modifications to their configurations.
- NRI plugins are runtime-agnostic and can be used with both containerd and CRI-O.



# NRI Plugin Operation Mechanism



- **Operation Mechanism**
  - The NRI plugin operates as an independent process, communicating with the runtime through the NRI socket and can be deployed as a Daemonset.
  - The NRI plugin, as a precompiled binary, is placed in a specified path and invoked by containerd (similar to CNI).
- **Hook Events**
  - Pod event: *RunPodSandbox(),StopPodSandbox(),RemovePodSandbox()*.
  - Container events: *CreateContainer(),StartContainer(),UpdateContainer(),StopContainer(),*
  - *RemoveContainer(), PostCreateContainer(),PostStartContainer(),PostUpdateContainer()*
- **Initiated Events**
  - *stub.UpdateContainer()*

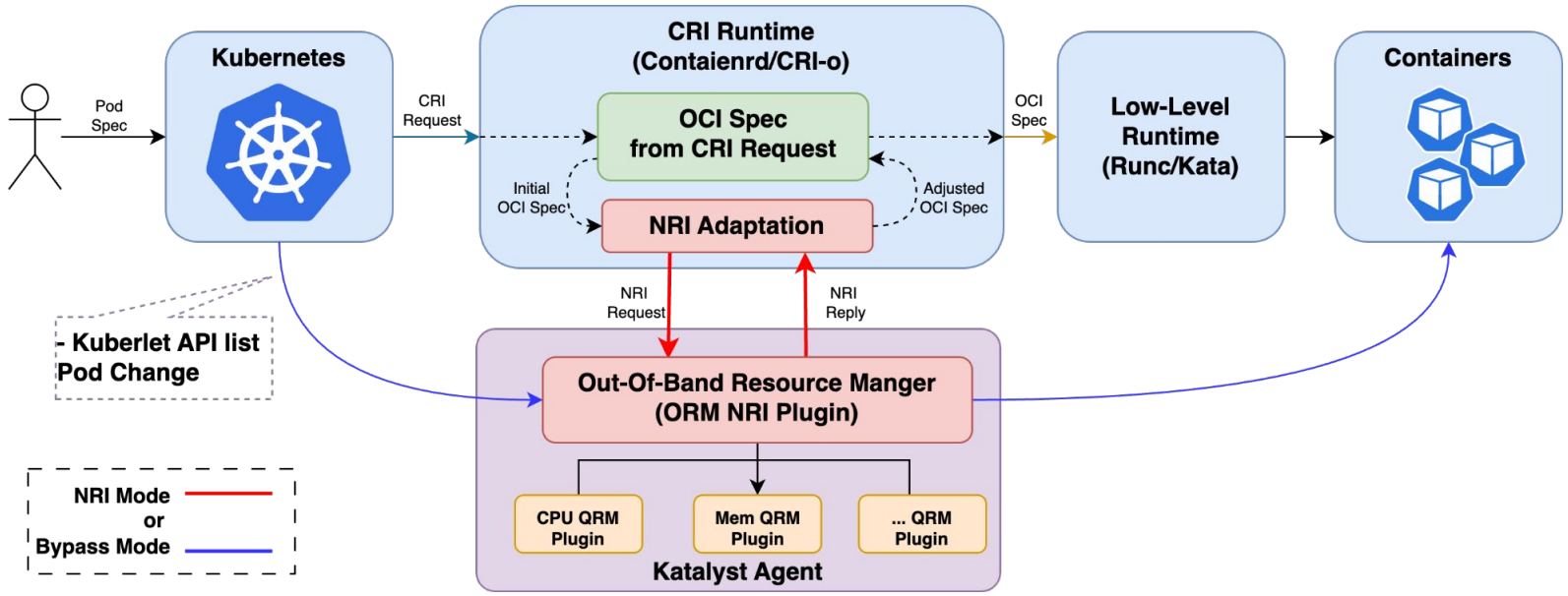


# 4 Application of NRI in Katalyst

# NRI Enhanced ORM in Katalyst

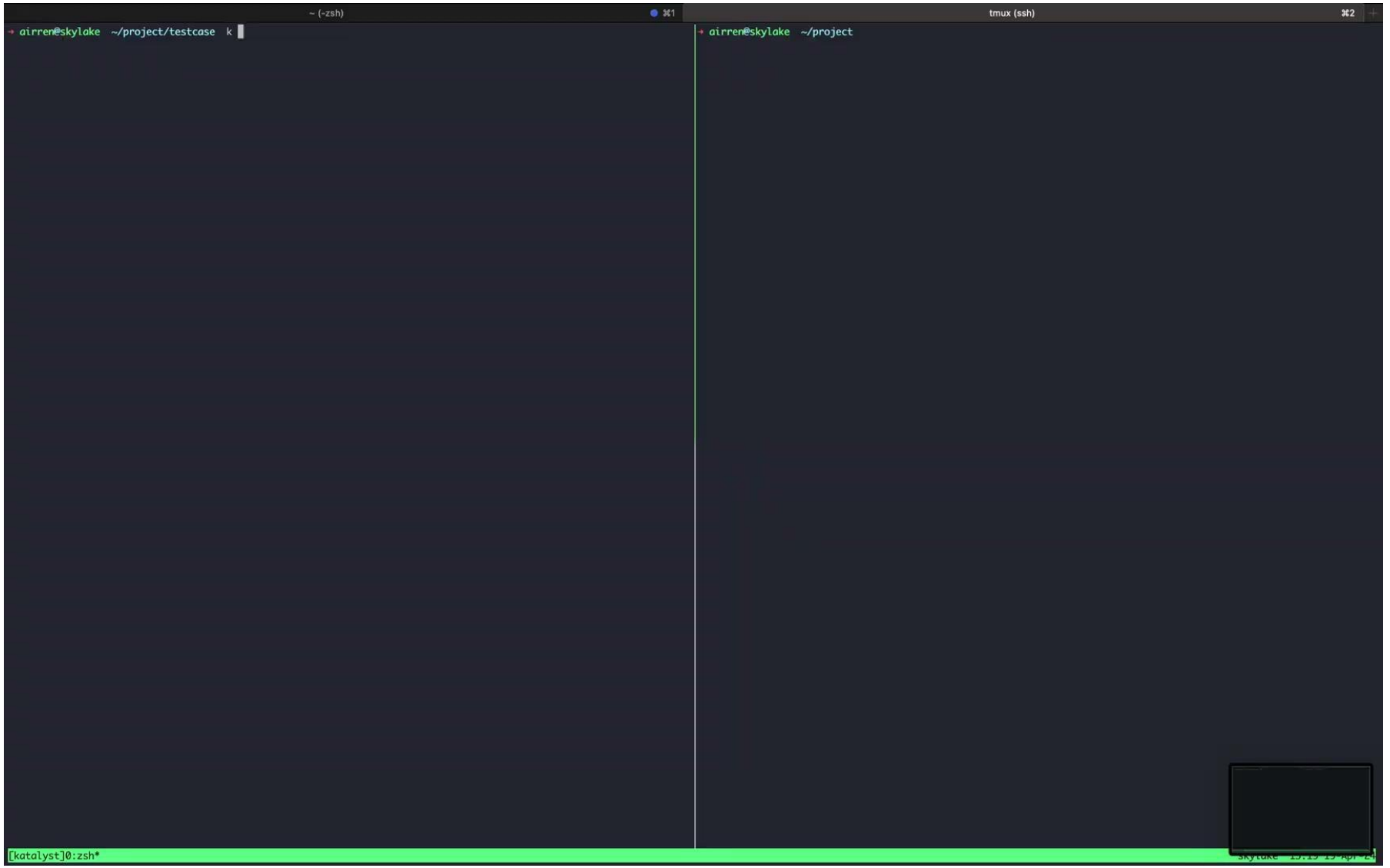


- The Katalyst Agent functions as an NRI plugin to receive Pod/Container events from containerd, such as *RunPodSandbox(), CreateContainer(), and RemovePodSandbox()*.
- Bypass mode coexists with NRI, adapting to older versions of the containerd environment.
- The QRM plugin mechanism is reused to reduce migration costs.
- The Reconcile mechanism updates container resources via NRI *UpdateContainer()*.



# Demo: NRI Enhanced ORM in Katalyst





# 5 Community

# Participating in the Communities

- **GitHub Repositories**
  - Katalyst: https://github.com/kubewharf/katalyst-core
  - NRI: https://github.com/containerd/nri
  - NRI Plugins: https://github.com/containers/nri-plugins
- **Katalyst Bi-weekly Community Meeting**
  - Thursday 19:30 GMT+8 (Asia/Shanghai)
  - Meeting notes and Agenda


- **He Cao**
  - Email: caohe.ch@bytedance.com
  - GitHub: @caohe
- **Qiang Ren**
  - Email: qiang.ren@intel.com
  - GitHub: @Airren

# KubeCon Booth

KubeCon CloudNativeCon China 2024 | THE LINUX FOUNDATION OPEN SOURCE SUMMIT | AI_dev

## Welcome to the **KubeWharf** booth at **S12**!


KubeCon
CloudNativeCon
THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
AI_dev
Open Source GenAI & ML Summit
China 2024
Thank you!