CXL Orchestration: Taming the Fabric

Grant Mackey, CTO Jackrabbit Labs

What I Want You To Take Away From This Talk

- Don’t wait for that shining city on a hill, CXL is useable today!
- Core stranding is a very real problem that CXL addresses
- Middleware makes it work and there needs to be more of it!

Kubernetes and CXL Fabric Attached Memory

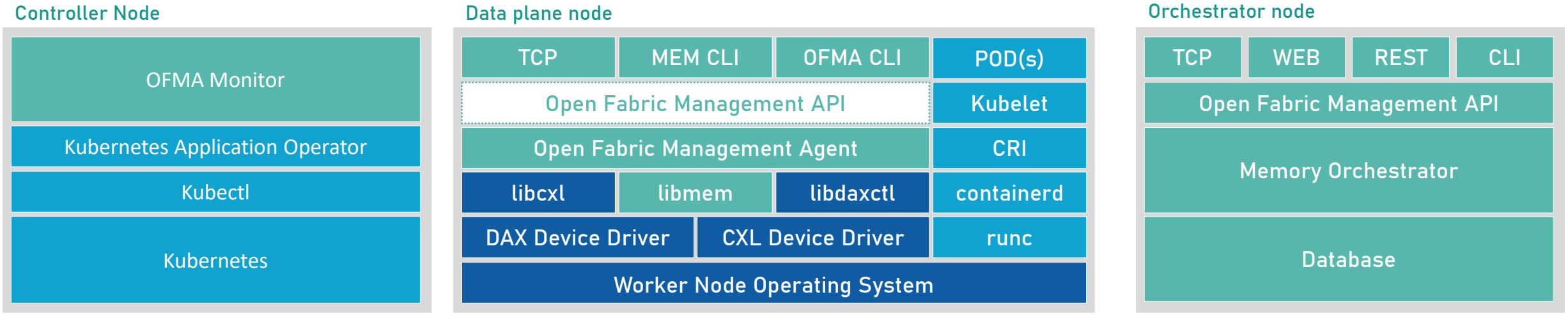
- K8s (Kubernetes) is a mature, datacenter(s) scale resource scheduler
  - Think Netflix, Uber, cloud providers that run or let you run containers, etc.
- For this talk, think of CXL Fabric Attached Memory as composable
- Now I’m going to talk about
  - why this comes off the rails for systems like K8s and others today
  - How you go about doing it
  - Why it has value

Kubernetes and Composable Memory

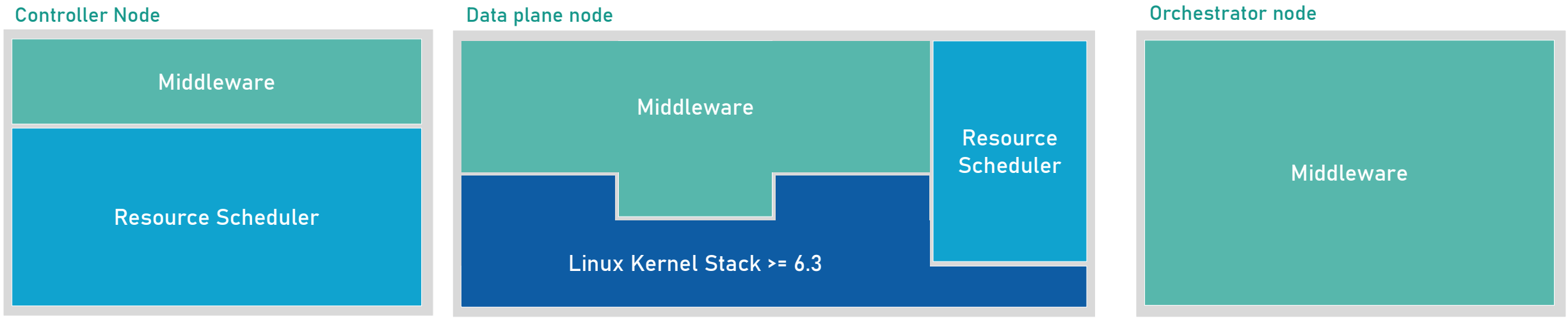
- In K8s and other resource schedulers, certain resources aren't 'supposed' to change.
- Modifying K8s to understand composable memory is a 'no' for now
- So what to do?
  - "Do no harm"
  - Integrate with the k8's lifecycle so the solution is robust
  - Use existing k8's NRI[1] to expose CXL memory as a resource type

1. Node resource interface https://github.com/containerd/nri

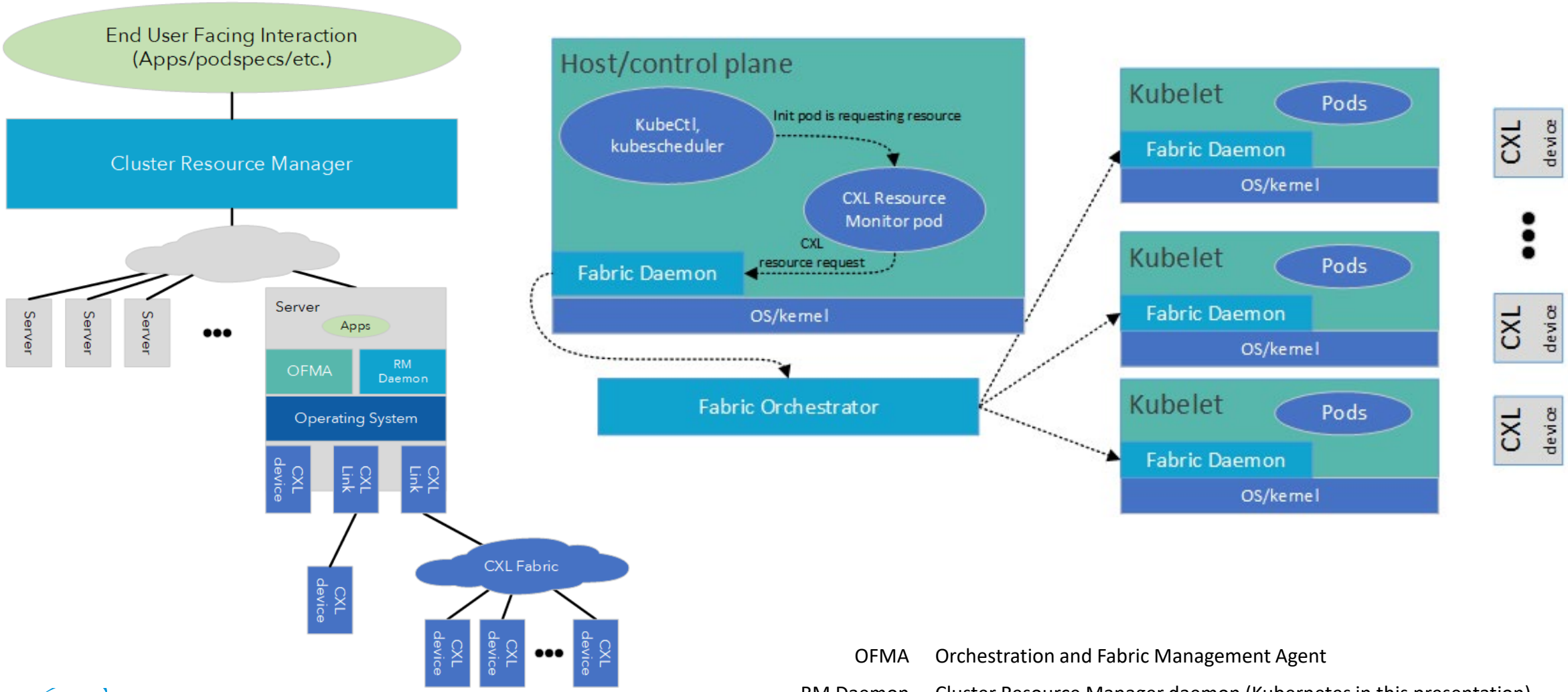
CXL Fabric-Attached Memory Stack



CXL Fabric-Attached Memory Stack



Composable Memory and Kubernetes

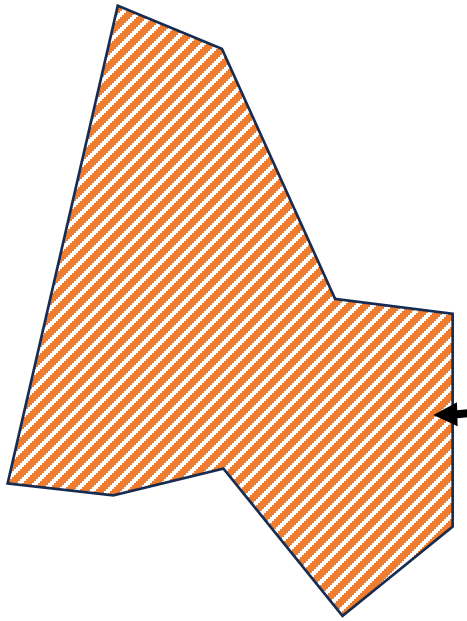


OFMA Orchestration and Fabric Management Agent

RM Daemon Cluster Resource Manager daemon (Kubernetes in this presentation)

Why Would Anyone Want This?

Why Would Anyone Want This?



This!!!

256GB/s per socket of memory bandwidth

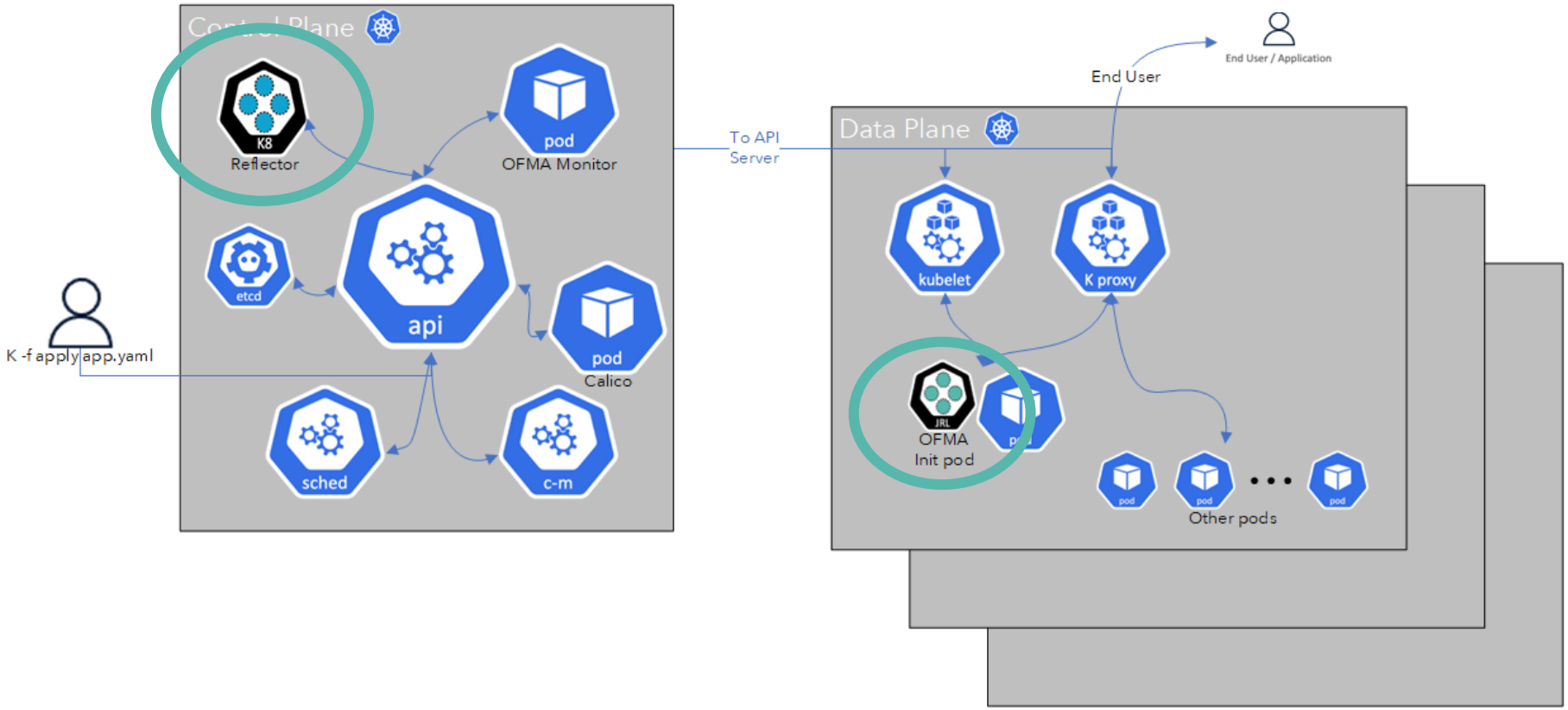
Apps that care about this

- Web workloads
  - Webservers, ruby, go, etc.
- Cloud Microservices
  - Lambda services, streaming media, etc.
- Databases that look like TPC-H
- Data Science workloads
  - Spark, Java, pandas, etc.
- KV stores
  - Memcache, Redis, etc.

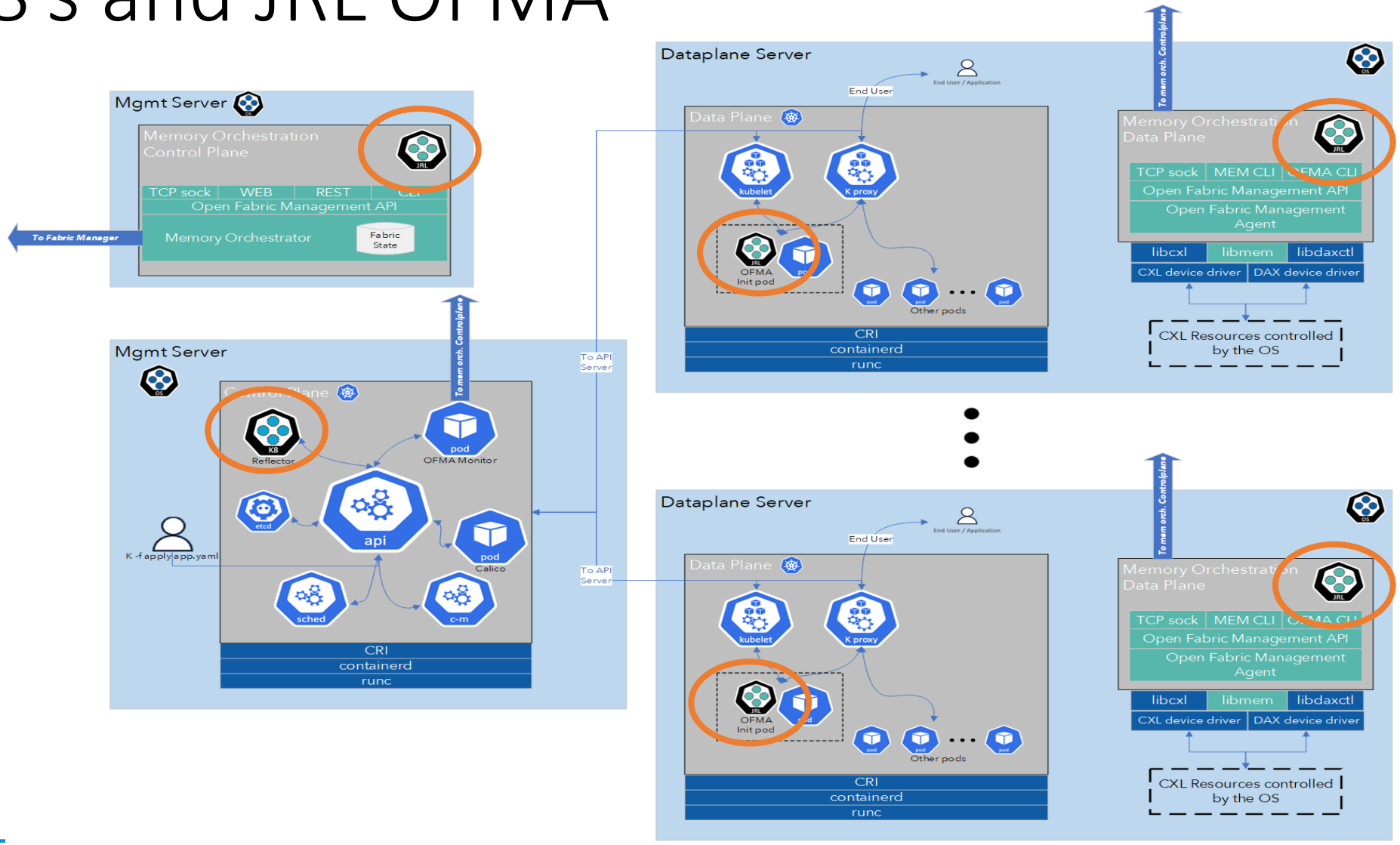
Call to Action

JACKRABBIT LABS

JRL K8's integration



K8's and JRL OFMA



Published Results from Others

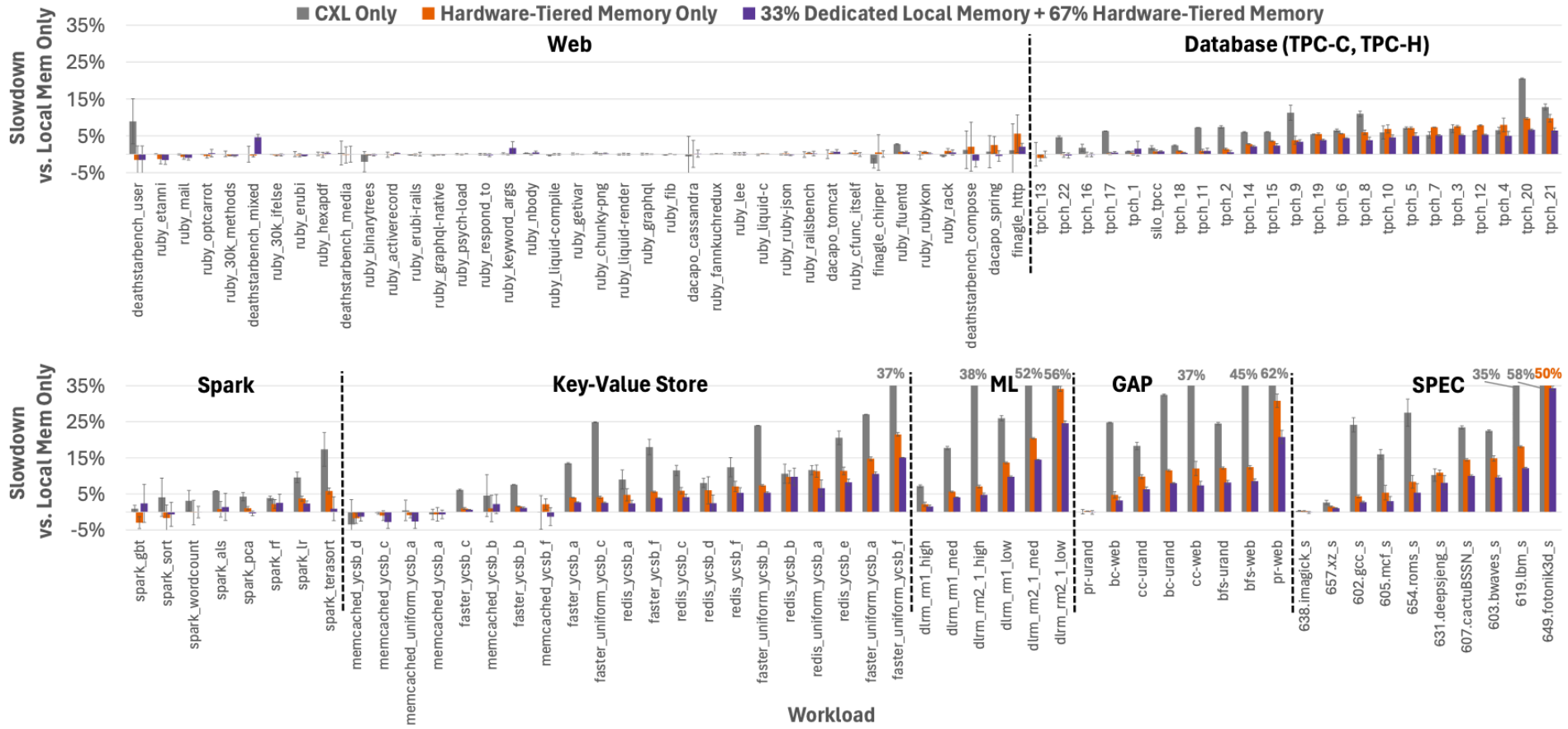


Figure 5: Slowdowns of 115 workloads when using only CXL memory, 100% hardware-tiered memory, or a mixed mode with 33% dedicated memory and 67% hardware-tiered memory. The error bars represent the standard deviations of slowdowns across three runs.

Managing Memory Tiers with CXL in Virtualized Environments, Zhong et. al. OSDI 2024