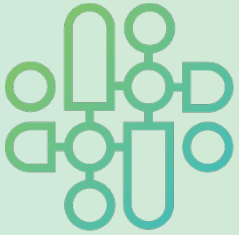


SC23
Denver, CO | i am hpc.

Survey of adaptive containerization architectures for HPC

Nina Mujkanovic
Tiziano Müller
HPE HPC/AI EMEA Research Lab

Juan Durillo
Nicolay Hammer
Leibniz Supercomputing Center (LRZ)

OUTLINE

- Background

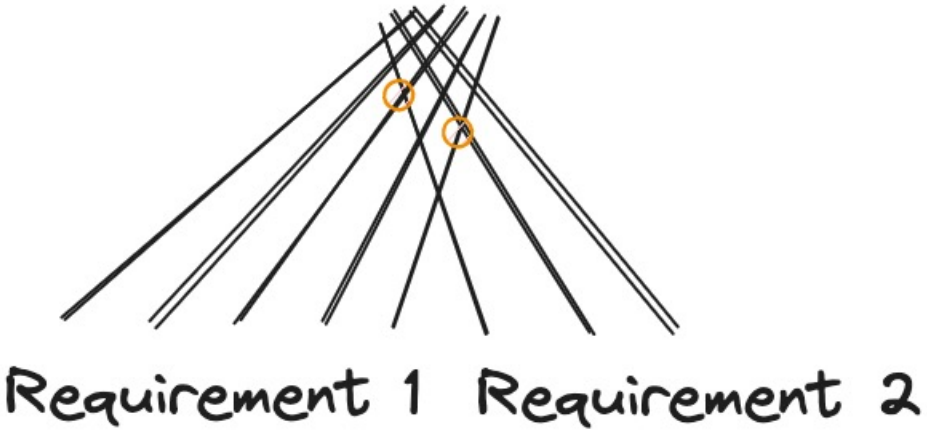
- Container Engines
- Container Registries
- Kubernetes WLM Integration

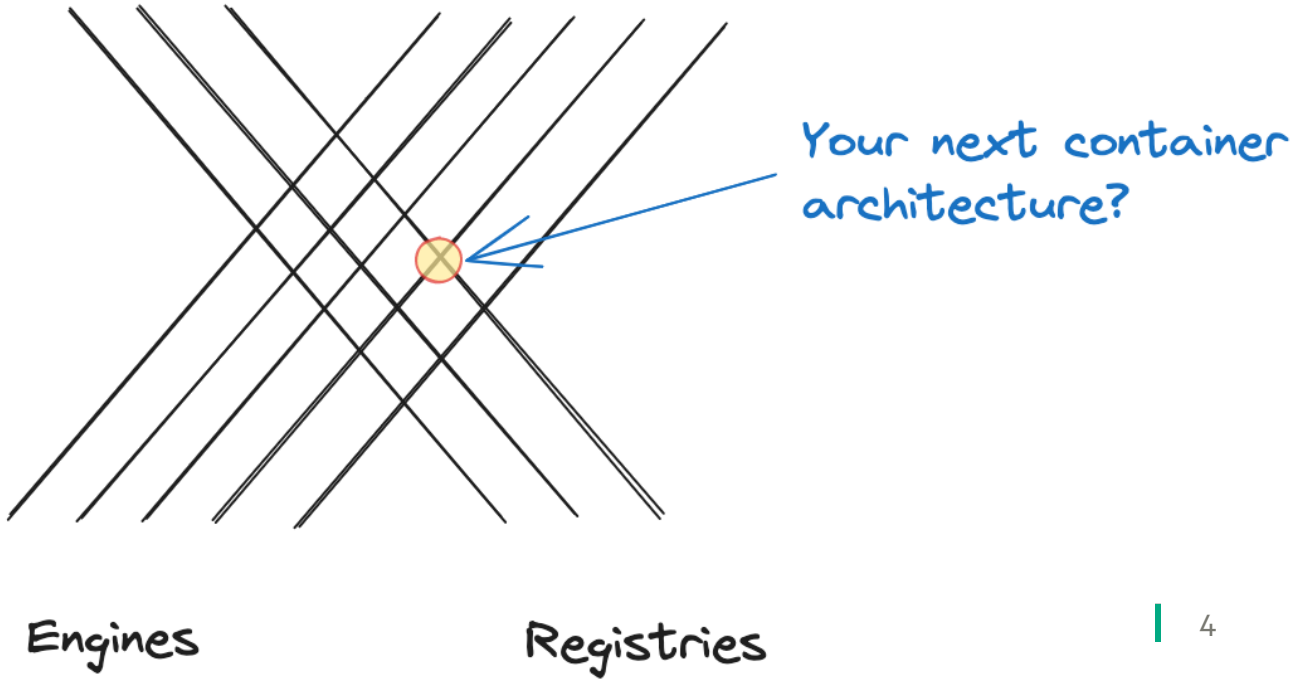
- Outlook

AQTIVATE EJD UCL CDT

ADAPTIVE ARCHITECTURES FOR HPC

- Consider user, system, and site-specific needs (risk management)
- Choose **best adapted** tool for each component
- Merge into an integrated workflow
- Accelerate deployment of applications and workflows using containers
- Integrate **container engines, registries, orchestration tools** to deliver full workflow capabilities





HPC REQUIREMENTS

- No privilege escalation – rootless imperative
- Strict container isolation on shared hardware…
- … but access to interconnects and accelerators, custom inter-node communication
- Filesystem access ….
- … and shared filesystems
- Target architecture optimization …
- … container portability
- Usually no continuously run services but may require scanning images as due diligence

CONTAINER ENGINES

CONTAINER ENGINES

- 9 Container engines
- 26 criteria

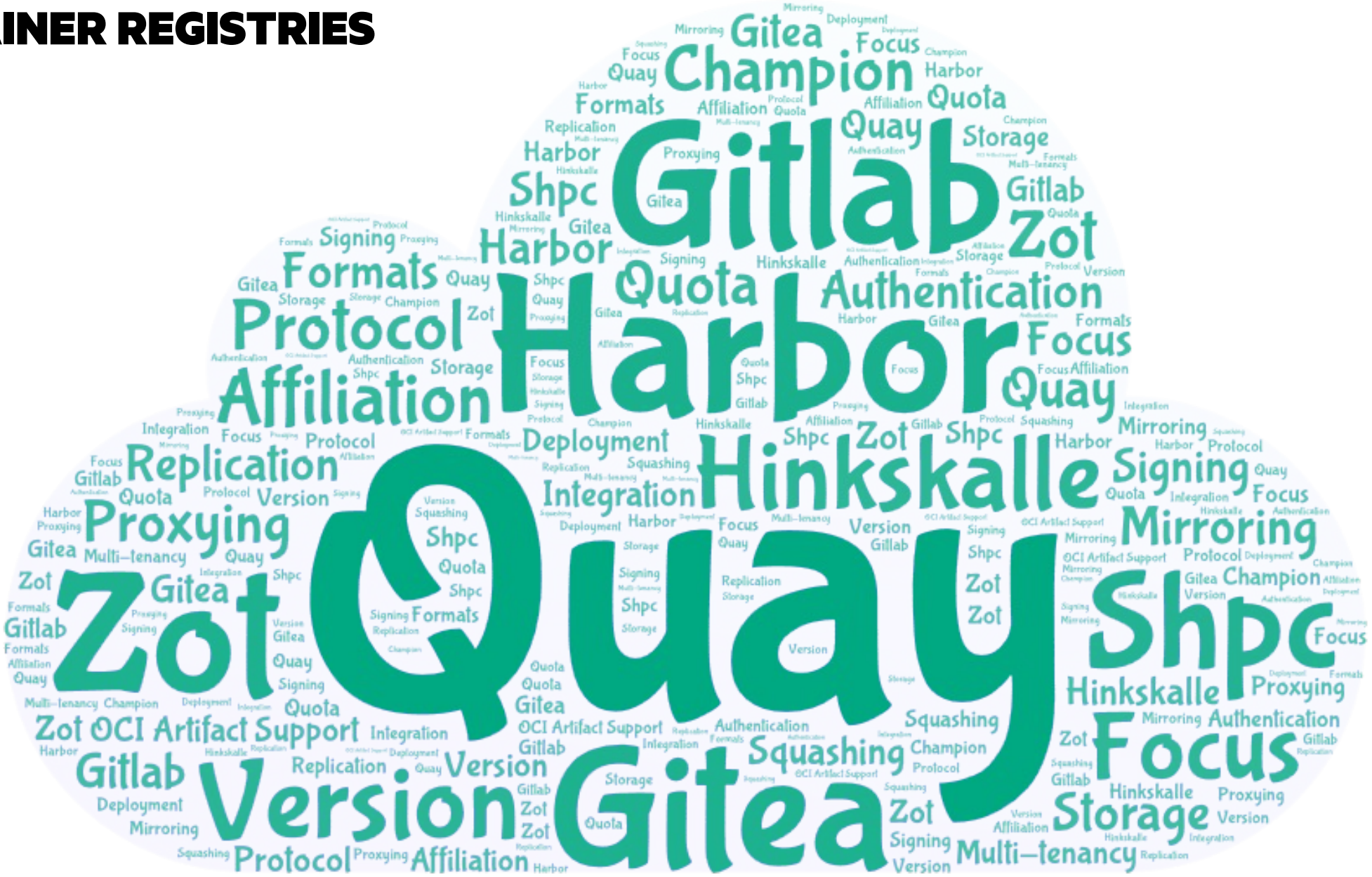
Container Solution	Docker	podman	podman-hpc	Shifter	Sarus	Charliecloud	Apptainer
Version	v20.10.21 (Oct. 25, 2022)	v4.3.1 (Nov. 10, 2022)	Git f0e7212 (Nov. 16, 2022)	Git 0784ae5 (Oct. 22, 2022)	v1.5.1 (July 11, 2022)	v0.29 (Aug. 5, 2022)	v1.1.3 (Oct. 25, 2
Champion	Docker	RedHat/IBM	NERSC	NERSC	CSCS	LANL	LLNL
Affiliation	Docker	Kubernetes					Linux Foundatior
Engine	Docker	podman	podman	Shifter	Sarus	Charliecloud	Singularity
Runtime	runC/crun	crun/runC/Crio-O	crun/runC/Crio-O	Shifter	runC/crun	Charliecloud	runC/crun
Engine Implementation Language	Go	Go	Python, C	C	C++	C	Go
Supports Rootless	yes	yes	yes	yes	yes	yes	yes
Rootless Implementation	UserNS	UserNS	UserNS	UserNS	UserNS	UserNS	UserNS, fakeroo
Rootless-FS Implementation	fuse-overlayfs	fuse-overlayfs	SquashFUSE + fuse-overlayfs	suid	suid	Dir, SquashFUSE	suid, fakeroot, (S
Container Monitor	per-machine (dockerd)	per-container (conmon)	per-container (conmon)	no	no	no	per-container (c
OCI Hook Support (in the runtime)	yes	yes	yes	no	yes	no	yes (manually, re
OCI Container Support	yes	yes	yes	yes (partial)	yes (partial)	yes (partial)	yes (partial)
Native Container Format	OCI	OCI, (SIF supported)	OCI, (SIF supported)	SquashFS	SquashFS	Dir, SquashFS	SIF
Transparent Format Conversion	-	-	yes	yes	yes	no	yes
(Transparent) Native Container Format Caching	-	-	yes	yes	yes	-	yes
Native Container Format Sharing	-	-	no	no	yes	no	yes
Namespacing Enabled on Execution	full	full	full/user and mount NS	user and mount NS	user and mount NS	user and mount NS	user and mount I others
Supported Signature Verification	Notary	GPG, sigstore	GPG, sigstore	-	-	-	GPG (only for SIF
Engine/runtime Support Encrypted Containers	no, extensions available	yes	yes	no	no	no	yes (SIF only, via
Builtin GPU-Enablement Support	no, possible via OCI hooks	no, possible via OCI hooks	yes	no	yes	no, manually	yes
General Accelerator Support	no, possible via OCI hooks	no, possible via OCI hooks	no, via OCI hooks or patch	no	via OCI hooks	no, manually	no
OS/MPI Library Hookup	no, possible via OCI hooks	no, possible via OCI hooks	yes	for MPICH	yes	no, manually	no, manually
WLM Integration	no	no	no	yes / SPANK plugin	partially via OCI hooks	no (no SPANK plugin release)	no
Contains Build Tool	yes	yes	yes	no	no	no	yes
Module System Integration	via shpc	via shpc	(via shpc)	no (shpc announced)	no (shpc announced)	no	via shpc
User Documentation	+++	+	N/A	+	++	+++	++
Admin Documentation	+	N/A	N/A	+	++	+	+
Questions on stackoverflow	yes	yes	no	no	no	no	yes
Number of Contributors	486	461	3	17	6	31	148
Source Code State and Documentation	+	++	(+)	++	+	++	+

CONTAINER ENGINES

CONTAINER ENGINES - SUMMARY

- Tripartite container tech space:
  - Cloud industry tools like Docker and podman
  - Singularity with its SIF format
  - New container integrations adhering to cloud industry standards
- HPC space previously trending to Singularity
- Podman easing shift with SIF support

CONTAINER REGISTRIES



CONTAINER REGISTRIES - SUMMARY

- CI/CD system registries offer direct build-storage systems but lack some features
- All support OCI except shpc (library API, SIF)
- Note: SIF images can be pushed to OCI registries, library API not a technical requirement
- HPC-centric candidates
  - Project Quay
  - Harbor

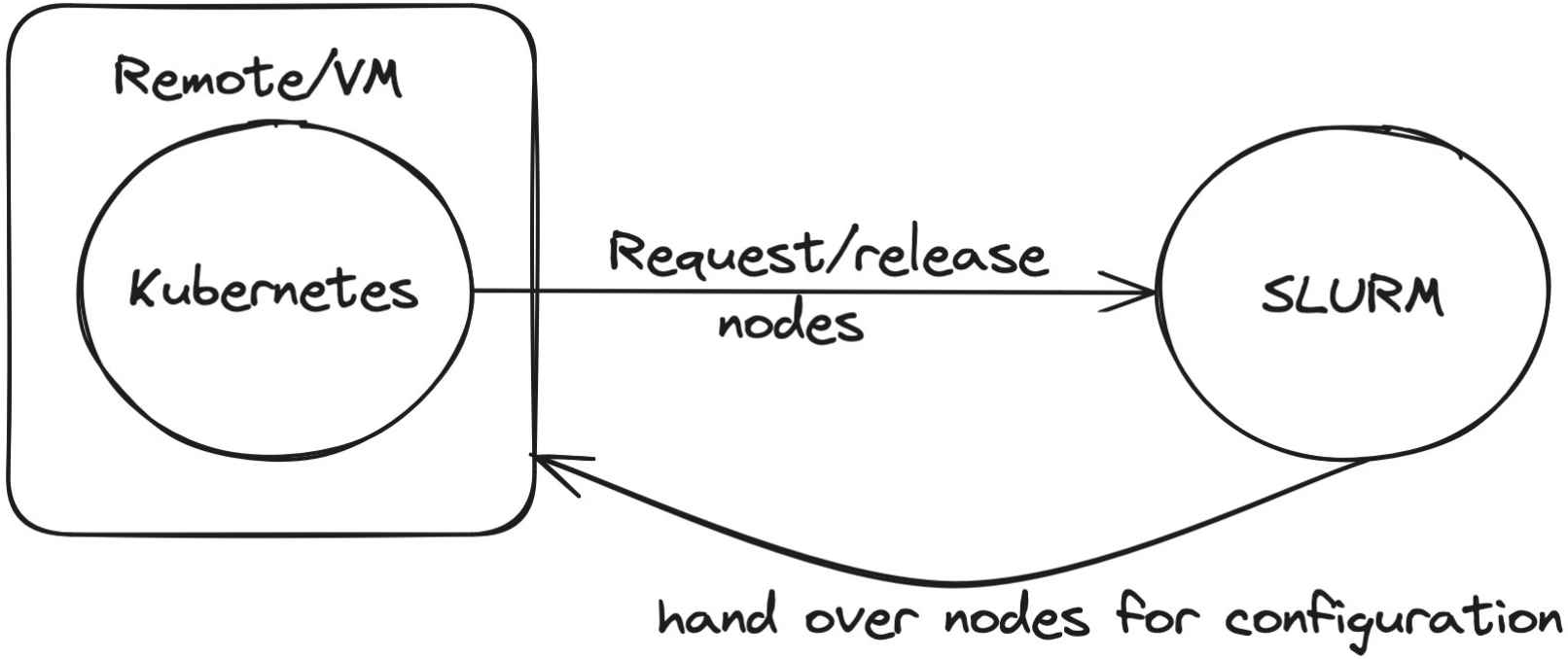
KUBERNETES WLM INTEGRATION

- Static partitioning = reduced utilisation and/or load imbalance
- Dynamic partitioning = cumbersome, slow, introduces disturbances, accounting difficulties

- Kubernetes supports namespace isolation and user mapping

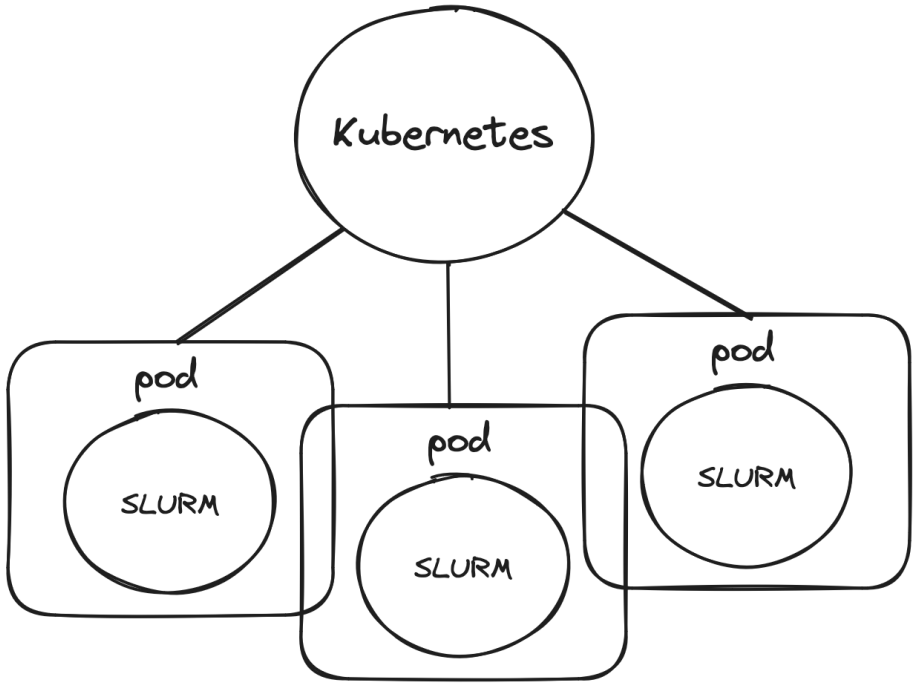
KUBERNETES WLM INTEGRATION

- On-Demand Reallocation of Compute Nodes
  - Kubernetes and WLM adjacent
  - k8s on separate hardware; Slurm takes requested nodes offline; k8s configures them; release to Slurm

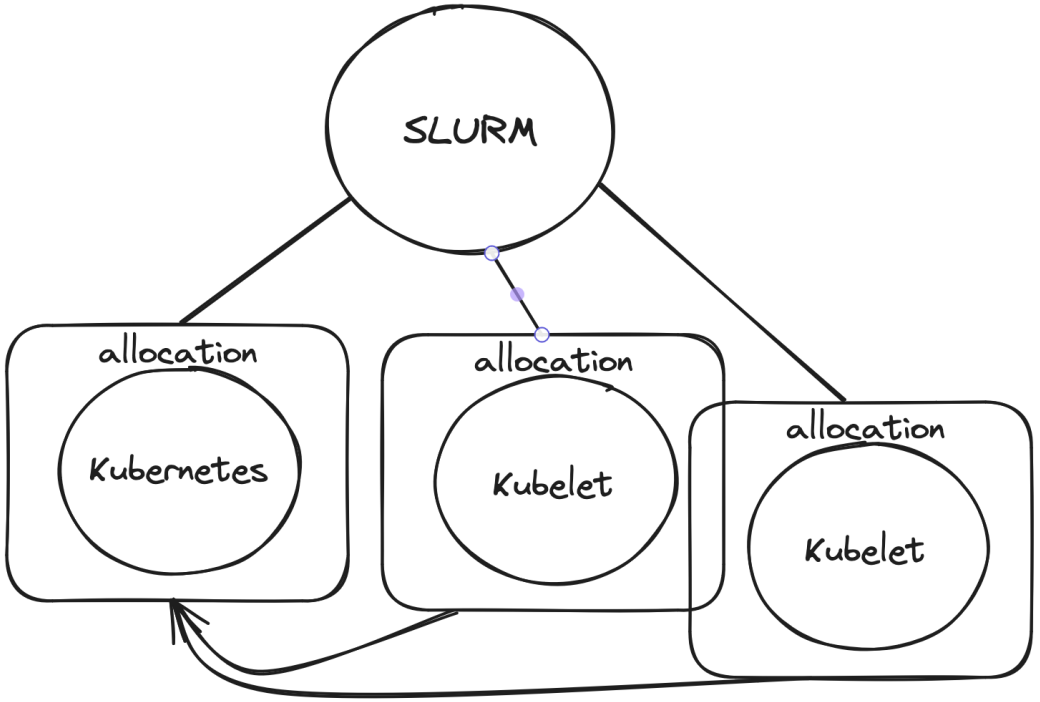


KUBERNETES WLM INTEGRATION

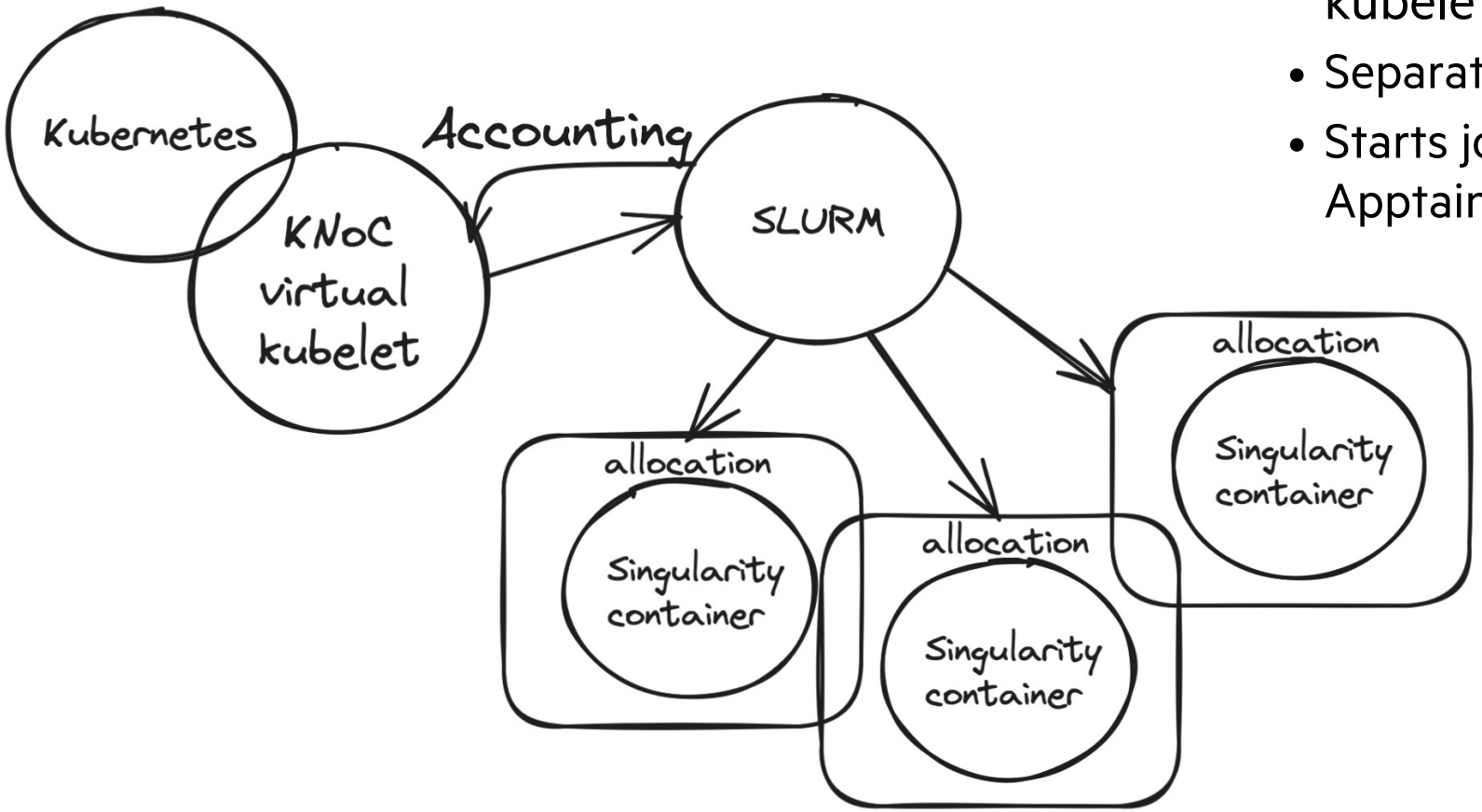
- WLM in Kubernetes
  - Simplest integration scenario
  - Does not enable running containerized workloads within the WLM
  - WLM acts as classical scheduler



- Kubernetes in WLM
  - Rootless k8s
  - First node runs minimal k8s instance; other nodes kubelets connecting back; network managed by WLM

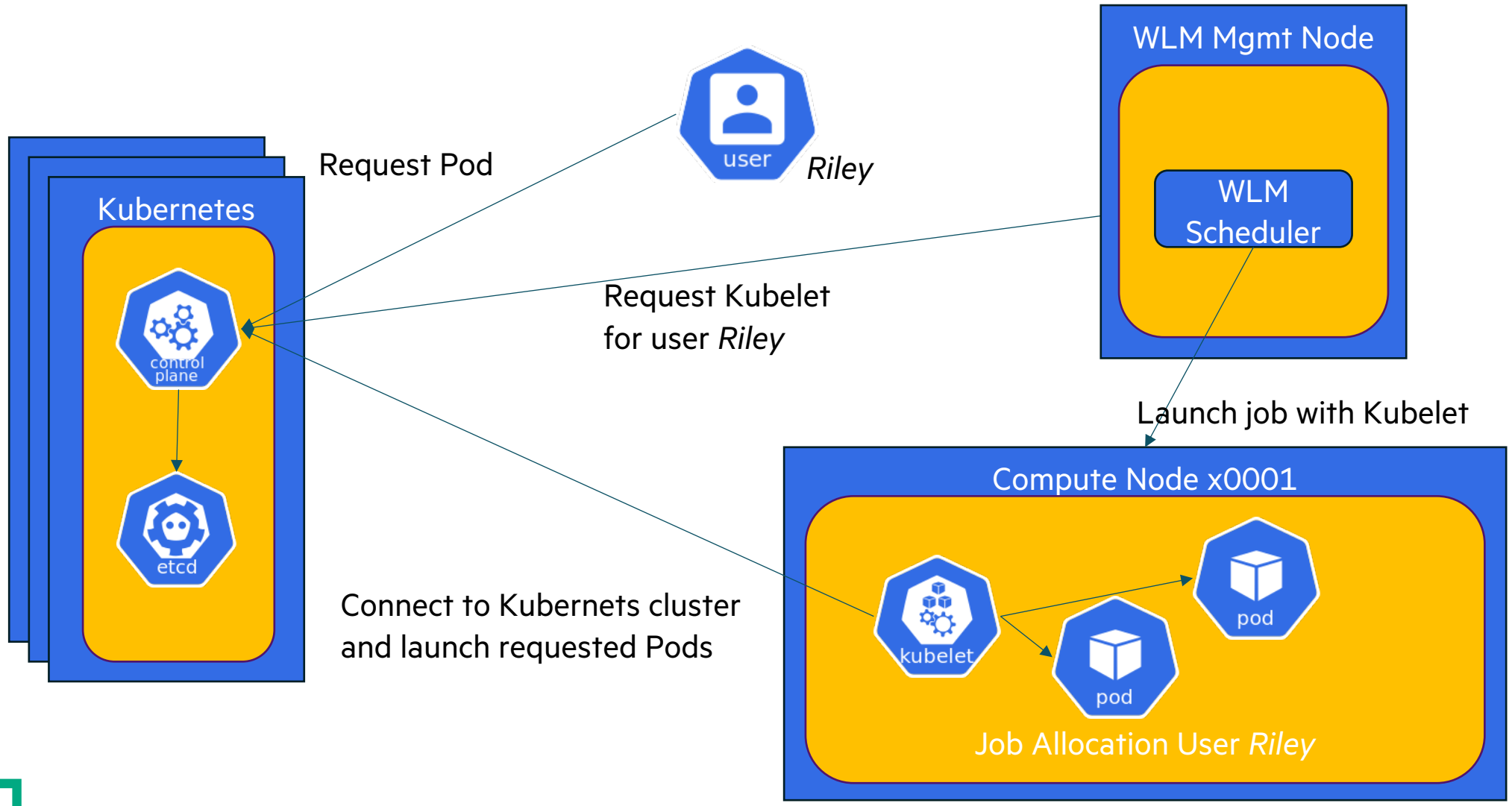


KUBERNETES WLM INTEGRATION



- Bridged Kubernetes and WLM
  - KNoC [1] – virtual Kubernetes agent or kubelet
  - Separate service acts as a regular kubelet
  - Starts jobs as container by using e.g. Apptainer within WLM allocation

KUBERNETES AGENTS IN WLM ALLOCATION – PROOF OF CONCEPT

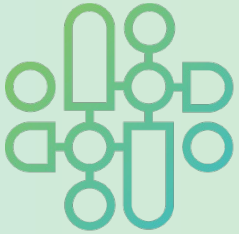


KUBERNETES WLM INTEGRATION - SUMMARY

- WLM in Kubernetes
  - isolation for multitenancy
  - no accounting via WLM, performance bottlenecks
- Kubernetes in WLM
  - isolation of k8s clusters
  - long startup times, allocation required before submitting to Kubernetes
- Most satisfactory solutions: Kubernetes agent in WLM allocation and KNoC
- Note: secure multi-tenancy and transparent scheduling of multi-node pods remains challenging

OUTLOOK

- Guidelines for labeling of containers for metadata-aware indexing, lookup, selection
- Federated container registry architectures incorporating caching, mirroring, proxying
- Secure containers and enclaves
- User guidance through selection of best matching container and optimal runtime parameters
- Automatic container optimization, build, storage, retrieval
- Site policies for BYOContainer



SC23

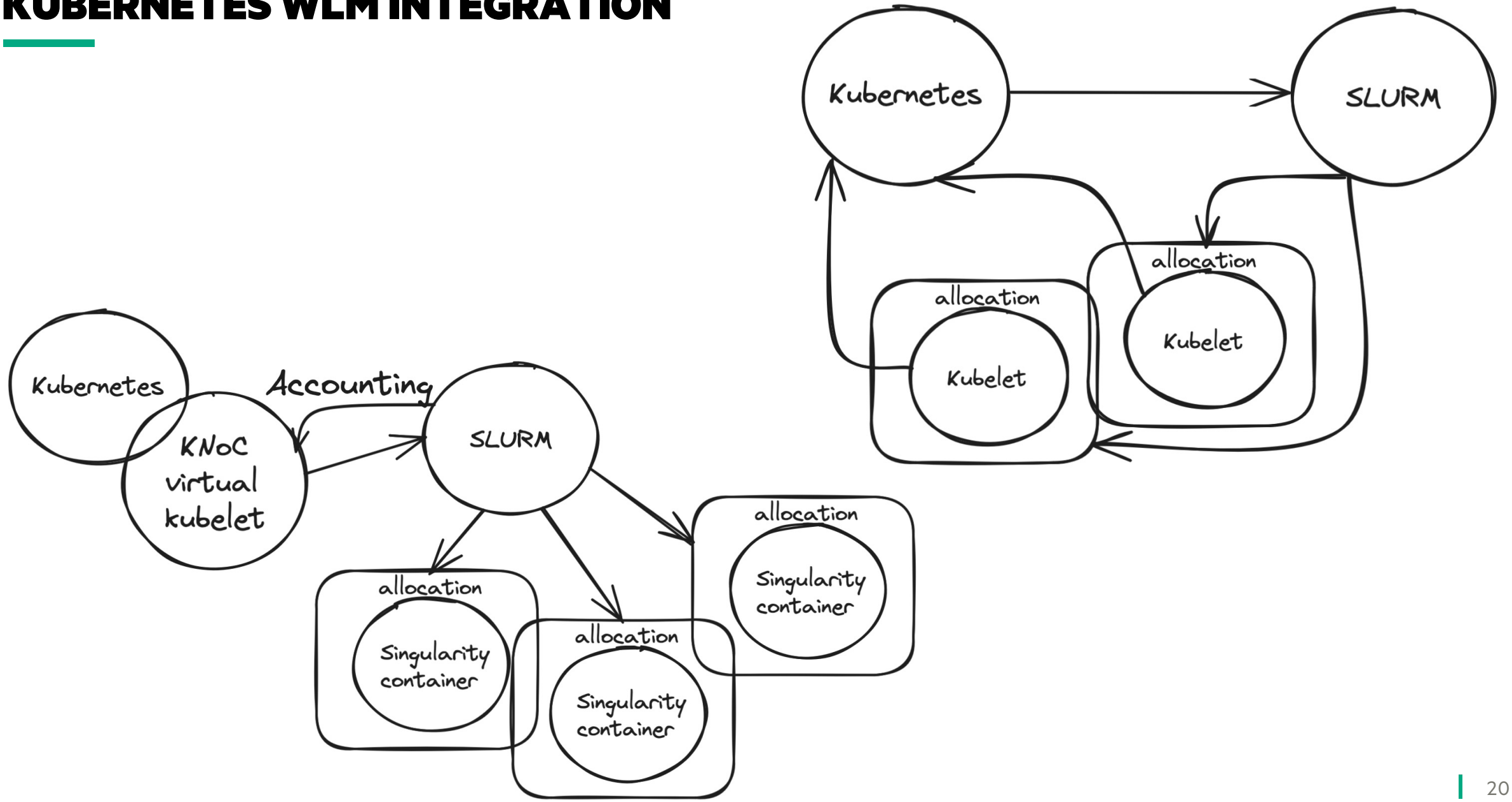
Denver, CO | i am hpc.

Thank you!

More questions? Write me! nina.mujkanovic@hpe.com

We thank the Gauss Centre for Supercomputing for funding this project as part of an innovation partnership aimed at a next-generation GCS supercomputer at the Leibniz Supercomputing Centre. We also thank the European Commission for continued funding of research on this topic under the Horizon project OpenCUBE (GA-101092984).

KUBERNETES WLM INTEGRATION



BIBLIOGRAPHY

[1] Evangelos Maliaroudakis, Antony Chazapis, Alexandros Kanterakis, Manolis Marazakis, and Angelos Bilas. 2022. Interactive, Cloud-Native Workflows on HPC Using KNoC. In High Performance Computing.

ISC High Performance 2022 International Workshops (Lecture Notes in Computer Science), Hartwig Anzt, Amanda Bienz, Piotr Luszczek, and Marc Baboulin (Eds.).

Springer International Publishing, Cham, 221–232. https://doi.org/10.1007/978-3-031-23220-6_15