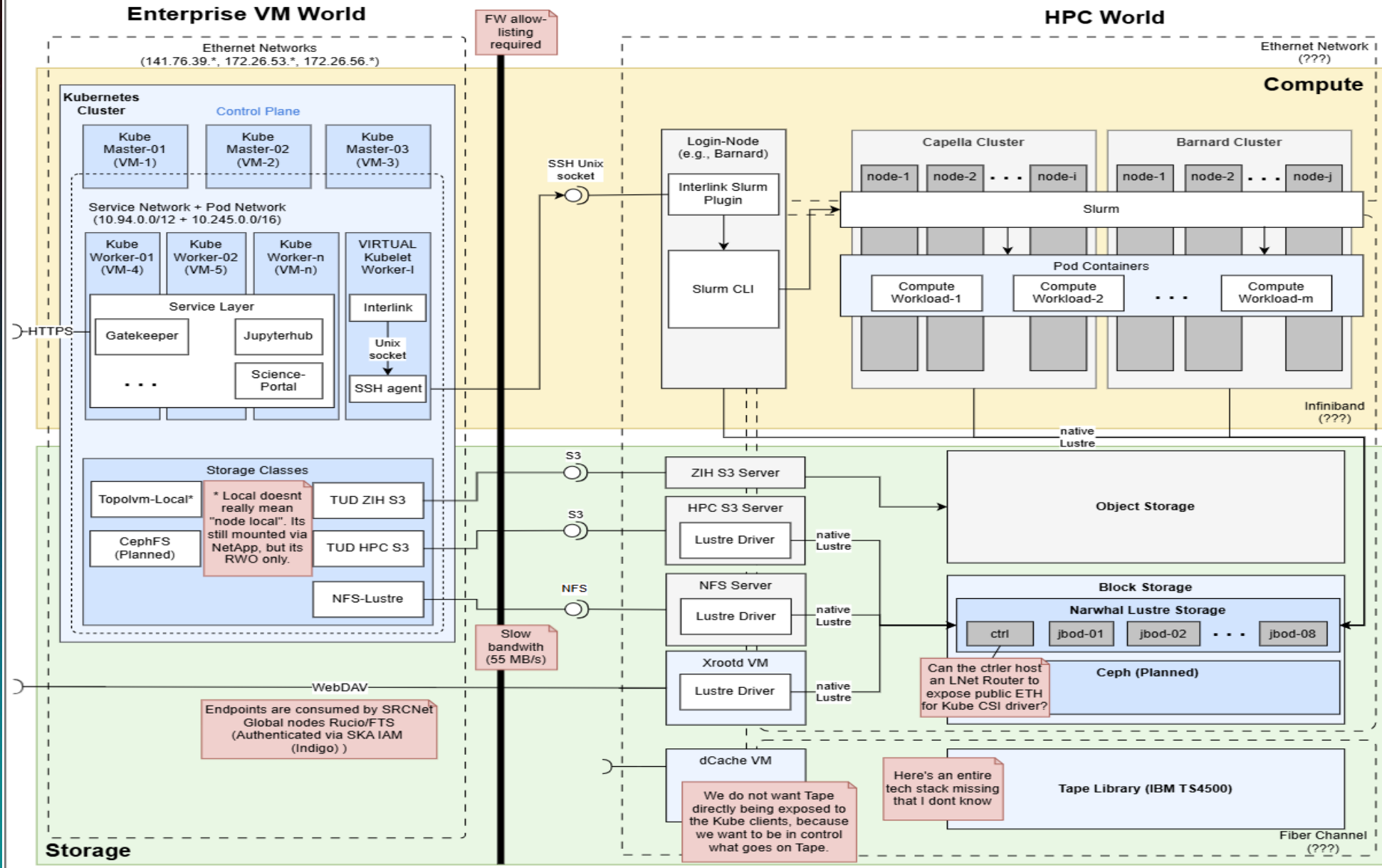


« Architectural overview of the dSRC node »

Marc Drobek

Team: Datascience

System Overview - Interlink Slurm SSH access



Landscape setup

/ 3

- Ideally 2 clusters:
 - INT: for testing
 - PROD: live system
- Currently:
 - One INT cluster for testing and to get to accreditation
- Deployment:
 - Via Flux GitOps pipeline
 - Still connected to DZA Gitlab -> needs to be moved over to SKA Gitlab

- Kube:
 - Infrastructure:
 - Calico, cert-manager, vault, external secrets, cnpg, flux, storage: topolvm, S3, NFS; traefik
 - SRCNet:
 - Gatekeeper, jupyterhub, canfar, preparedata, soda, cavern, posix-mapper, science-portal,
 - Monitoring:
 - Prometheus stack: Prometheus, Grafana, Loki, Alloy
 - Grafana alert integration to Mattermost channel for quick notifications
 - Service mgmt:
 - we install everything on INT cluster first to test
 - the idea is, that if the testing succeeds there, it will be rolled out to PROD cluster (which we do not have yet)
- Storage:
 - Xrootd for Rucio integration to global node
 - on Lustre storage (horribly bad Kube integration 😞)

Summary – Tech Challenges

(recap from Dec 2025)

/ 5

1. Is it better to have one single beefy Kube cluster or schedule workload outside?
2. Can we reliably and consistently schedule workload via Kube on an existing Slurm cluster?
3. What is the best solution to make the same data accessible inside Kube for a service/app and inside Slurm for heavy computation (without copying the data)?
 1. BlockStorage vs ObjectStorage vs NFS vs etc.
 2. Can we avoid data copies between “user areas” vs “raw data areas”
4. What will be the “winning” orchestration strategy of the astro algorithm/workload?
 1. Push via packaged container as “standalone” workload in Kube?
 2. Push via “direct code” through Software platform (e.g., Jupyterhub, Canfar, etc.)?
5. Ethernet (Kube) vs Infiniband (HPC/storage) -> how to translate workload?
6. And plenty plenty more ...

Thank you