

SPDK-CSI: Bring SPDK to Kubernetes

Yibo Cai
Arm

Agenda

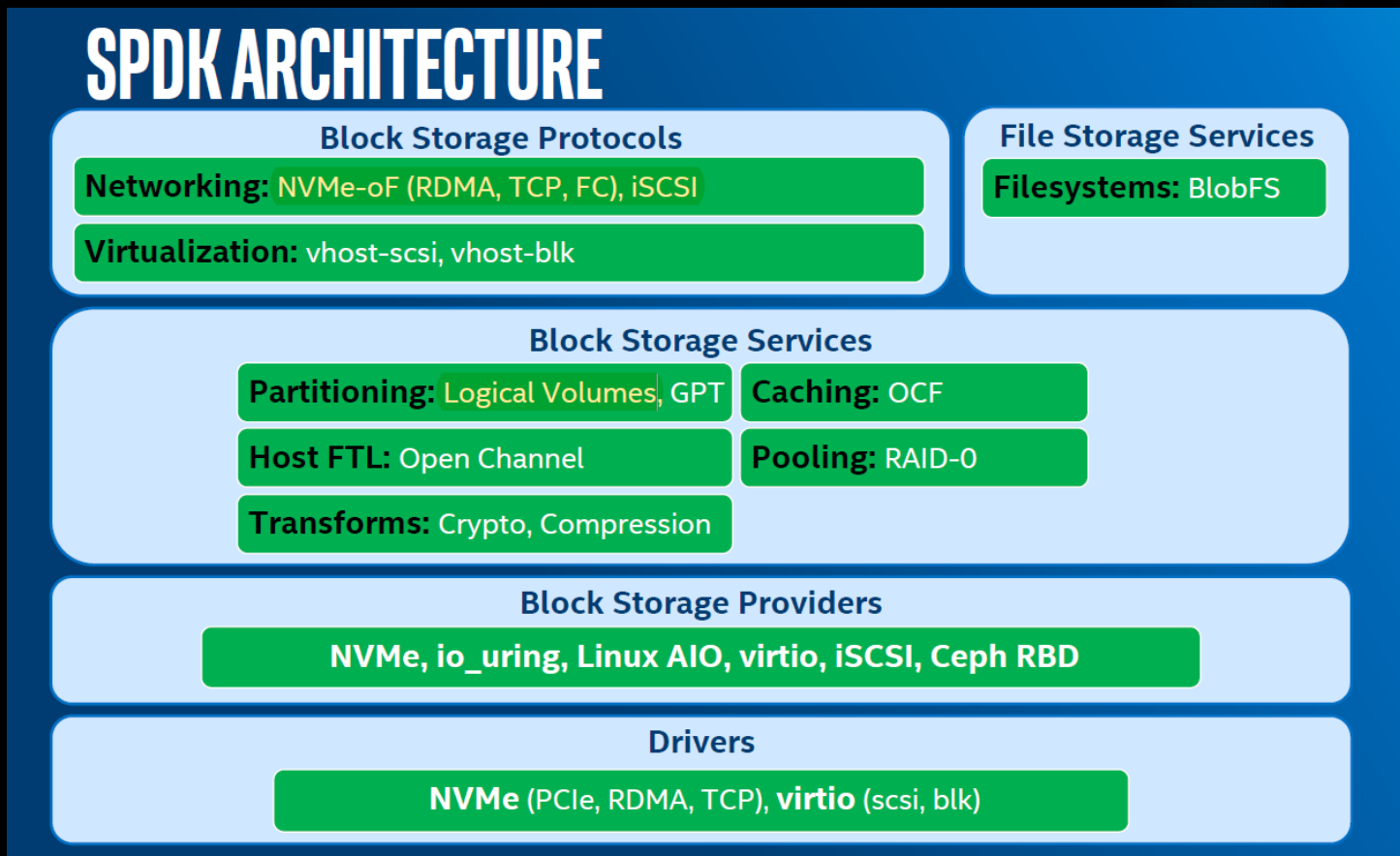
- SPDK Briefs
- Container Storage Interface (CSI)
 - CSI Internals
 - Kubernetes CSI development
- SPDK-CSI Implementation
- SPDK-CSI Community

SPDK

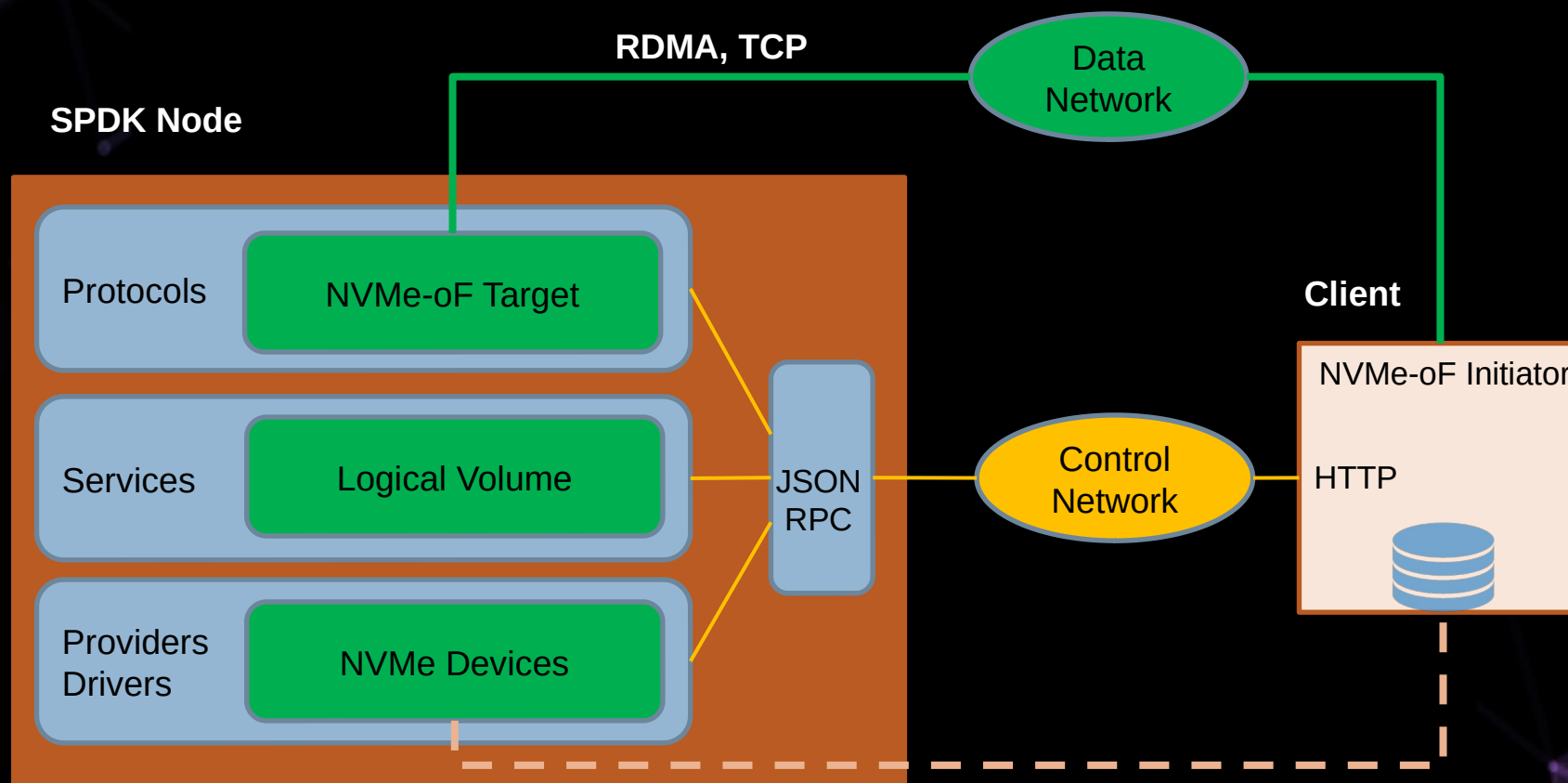


What is SPDK

- Quoted from <https://spdk.io/>
 - The Storage Performance Development Kit (SPDK) provides a set of tools and libraries for writing *high performance, scalable, user-mode* storage applications.
- Key techniques
 - Interact with hardware directly in user space
 - Polling data readiness instead of interrupt
 - No locks in I/O path



SPDK Network Storage



Container Storage Interface (CSI)



What is CSI

- Kubernetes volume driver: a brief history
 - In-Tree: storage driver coupled in Kubernetes code base.
 - Deprecated, legacy code will be removed.
 - FlexVolume: exec based API for volume plugins.
 - Hard to deploy and manage dependency.
 - Container Storage Interface (CSI)
 - Addresses pains of In-Tree and FlexVolume.
 - Standardizes storage system integration with Kubernetes.
 - [Kubernetes CSI Drivers List](#)

What is CSI – An Example

To use Ceph RBD (block device) in a K8s pod

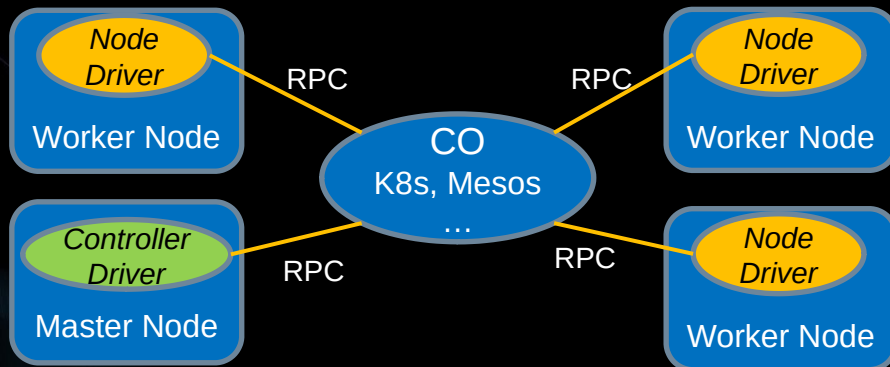
| No. | Steps | Run command at |
|------------|--|----------------------------------|
| App starts | | |
| 1 | Create RBD volume in Ceph cluster through Ceph API | Any host can access Ceph cluster |
| 2 | Mount RBD to host and container directory | Host where the pod runs |
| App stops | | |
| 3 | Unmount RBD directory | Host where the app runs |
| 4 | Delete RBD volume in Ceph cluster through Ceph API | Any host can access Ceph cluster |

How CSI automates the procedure

| What we need | In CSI Term |
|---|-------------------|
| [Step 1, 4] A storage driver to handle Ceph API and create/delete RBD on demand. It can run on any host which has access to Ceph cluster control plane. | Controller Driver |
| [Step 2, 3] A storage driver to (un)mount Ceph RBD volumes. It must run on all hosts where pods may be scheduled. | Node Driver |
| A protocol to define messages between K8s master and the plugins, so they can cooperate to finish the job. | RPC |

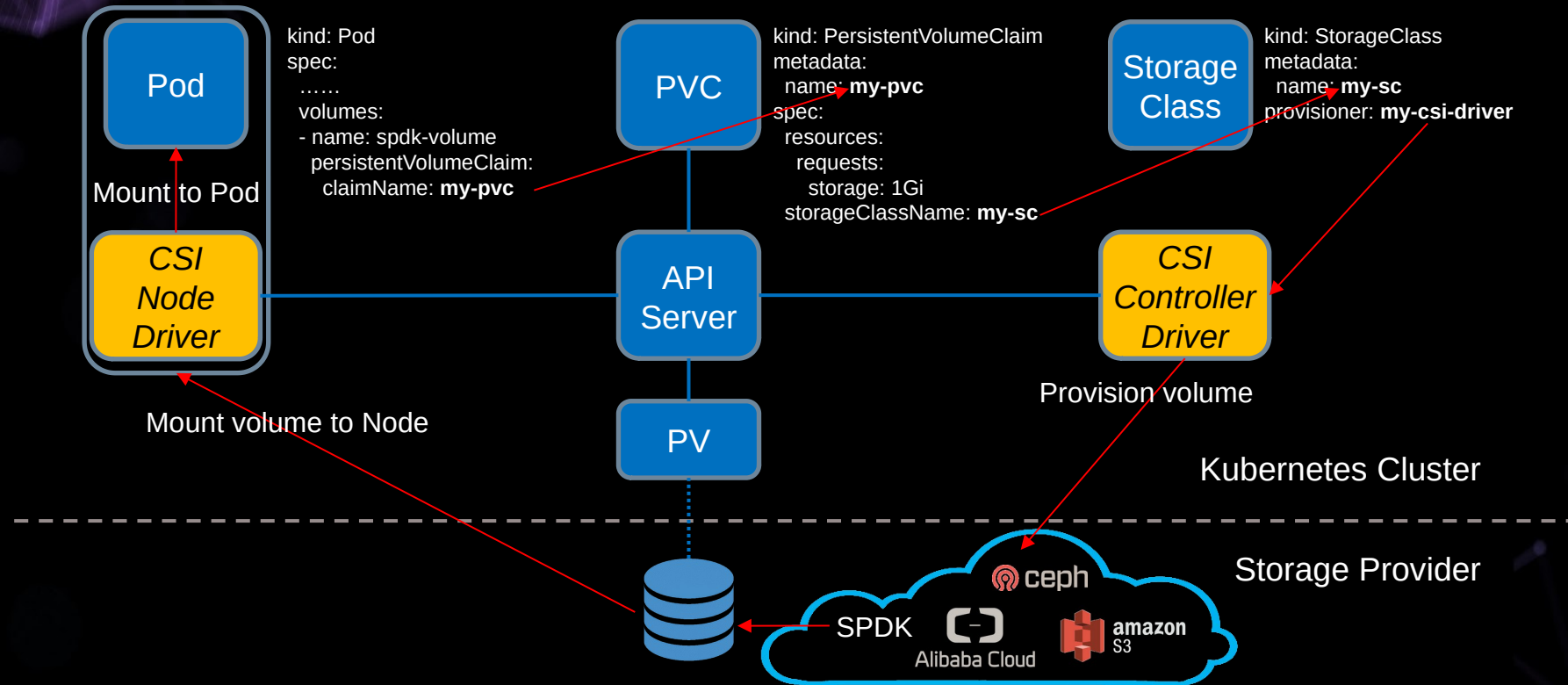
CSI Drivers and RPCs

- Controller Driver
 - Talk to Service Provider (SP) to create/delete volumes
- Node Driver
 - Mount/unmount remote volumes to local host



| RPC | Explains |
|-------------------------------|---|
| CO ↔ Controller Driver | |
| CreateVolume | Create a volume with specific parameters in storage provider |
| DeleteVolume | Revert creating |
| ControllerPublishVolume | Expose the volume to be accessible from worker node |
| ControllerUnpublishVolume | Revert publishing |
| CO ↔ Node Driver | |
| NodeStageVolume | Import remote volume and mount to worker node host |
| NodeUnstageVolume | Revert staging |
| NodePublishVolume | Bind mount host staging directory to container internal directory |
| NodeUnpublishVolume | Revert publishing |

Dynamic Volume Provisioning with CSI

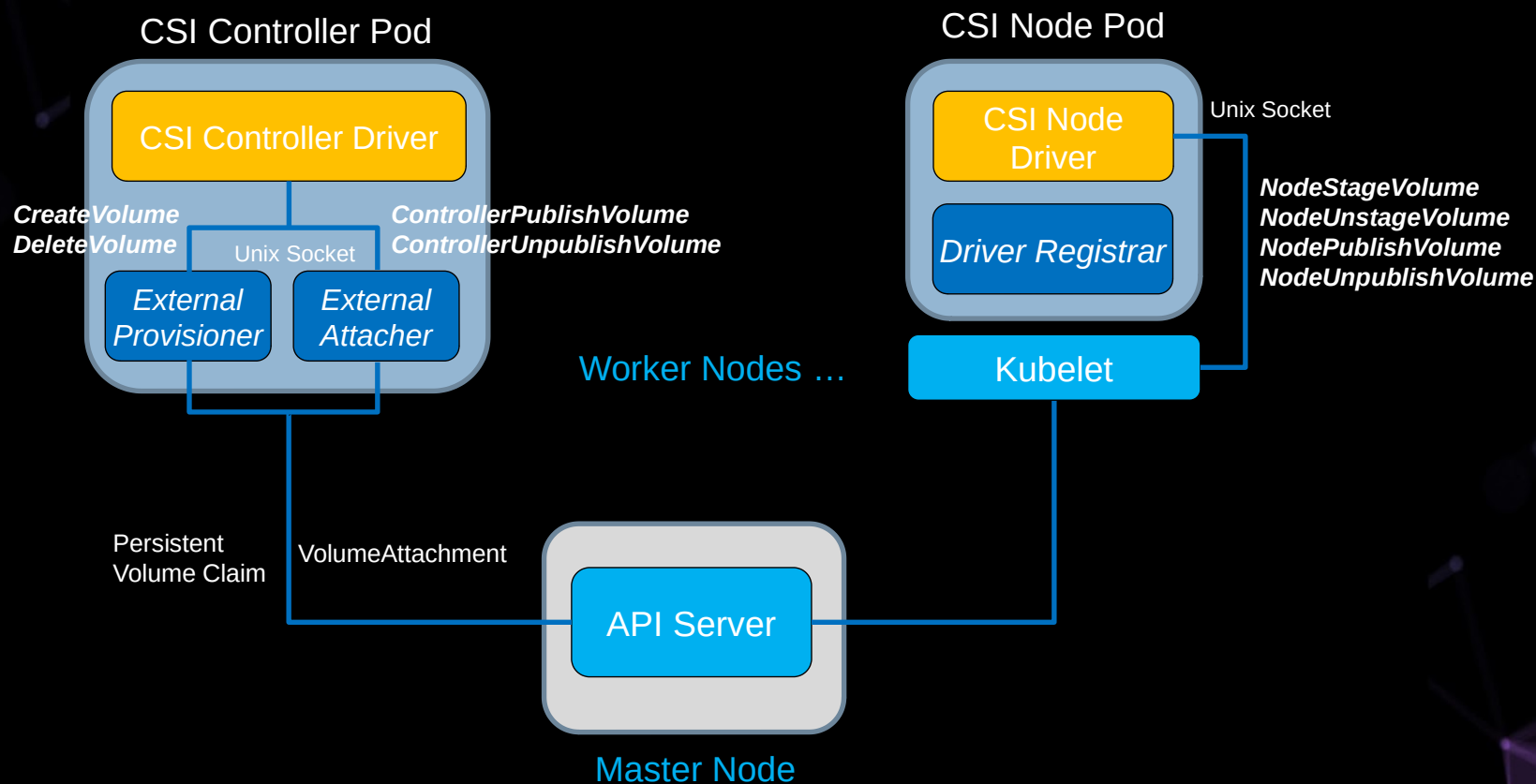


Kubernetes CSI Support

- Wrap Controller and Node driver in a single binary. Select functionality per command line.
- Deploy Controller driver as Deployment or StatefulSet
- Deploy Node driver as DaemonSet
 - Exactly one instance on each worker node
- Leverage CSI Sidecar containers to reduce boilerplate code

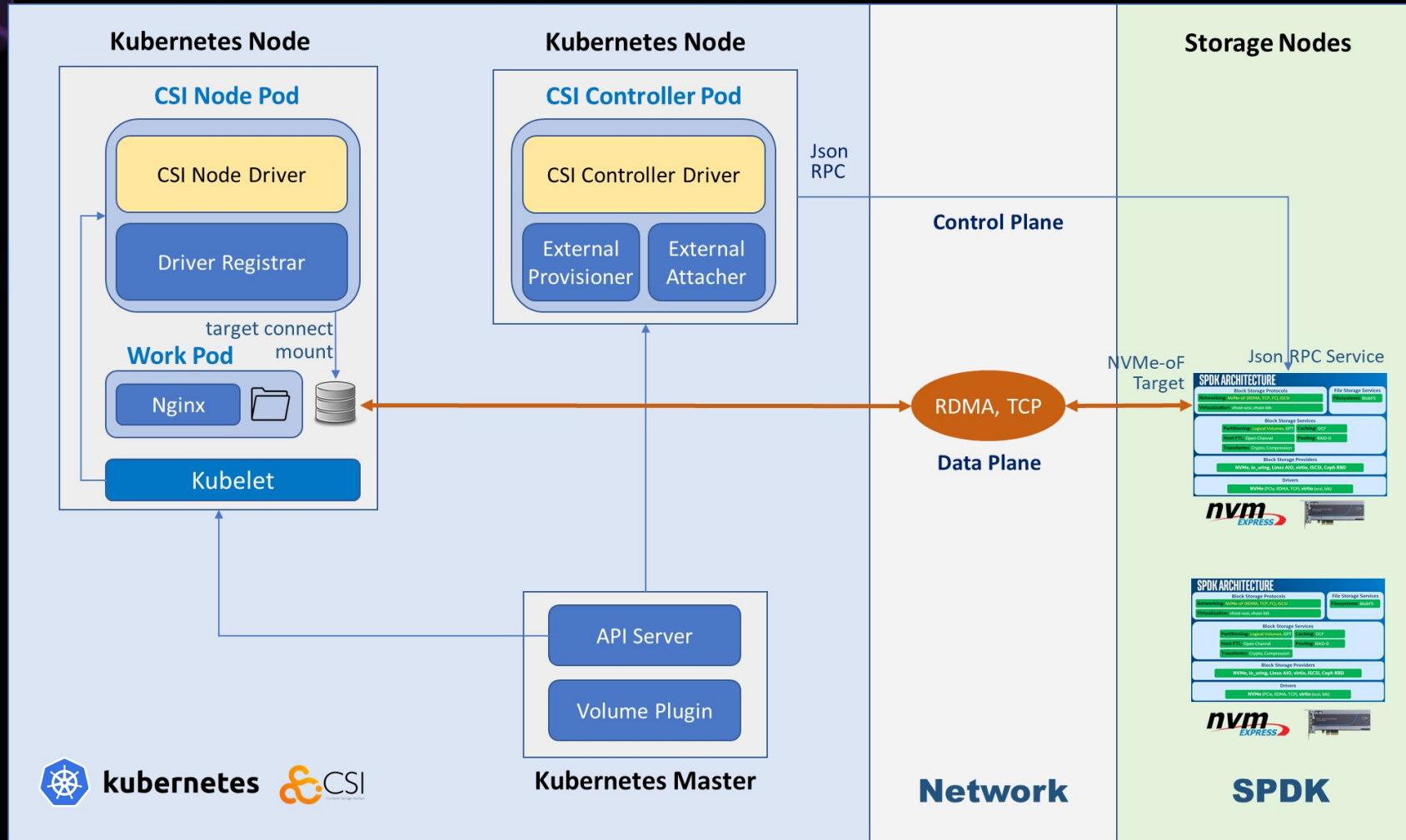
| Sidecar | Purpose |
|---------------------------------------|---|
| External Provisioner | Watches for PersistentVolumeClaim objects and triggers [Create Delete]Volume operations |
| External Attacher | Watches for VolumeAttachment objects and triggers Controller[Publish Unpublish]Volume operations |
| Node Driver Registrar | Registers the CSI driver with Kubelet to receive Node[Stage Unstage Publish Unpublish]Volume operations |
| | |

Kubernetes CSI Support



SPDK-CSI Implementation

Overview



Controller Driver

Controller configures SPDK network target through JSON-RPC

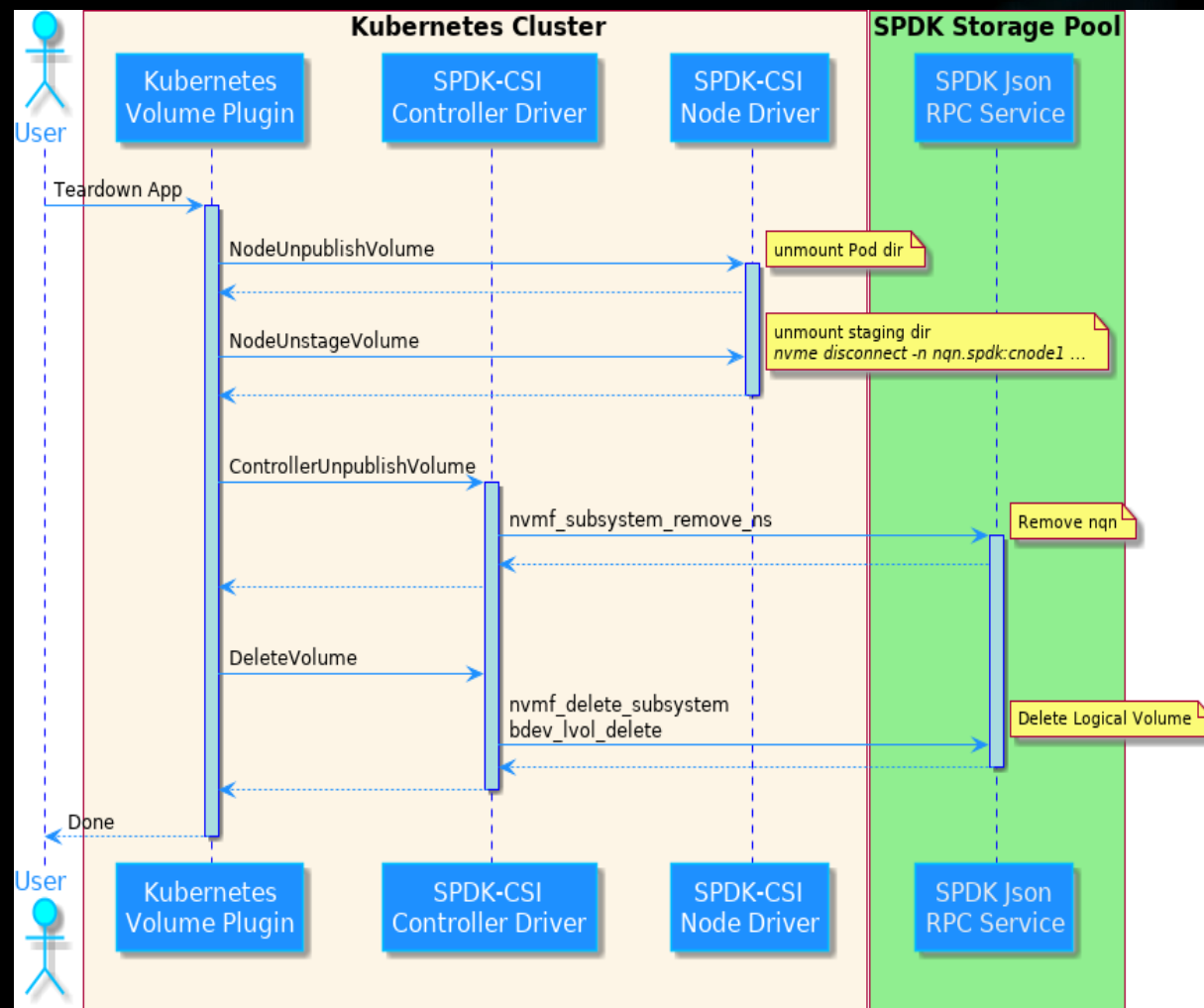
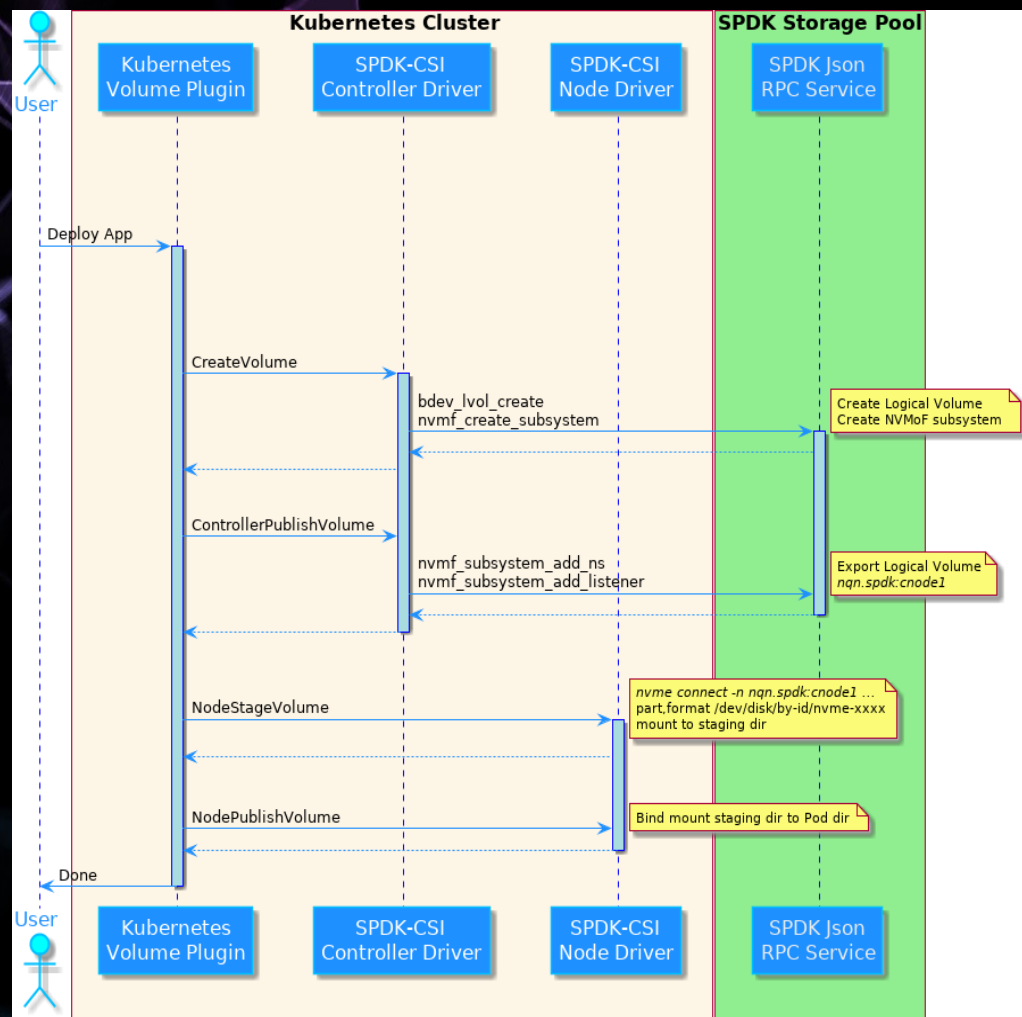
| CSI RPC | SPDK JSON-RPC (NVMf) | SPDK JSON-RPC (iSCSI) |
|---------------------------|--|---|
| CreateVolume | bdev_lvol_create | bdev_lvol_create |
| DeleteVolume | bdev_lvol_delete | bdev_lvol_delete |
| ControllerPublishVolume | nvmf_subsystem_add_ns nvmf_subsystem_add_listener | iscsi_create_portal_group iscsi_create_initiator_group iscsi_create_target_node |
| ControllerUnpublishVolume | nvmf_subsystem_remove_ns | iscsi_delete_target_node |

Node Driver

- Node connects to SPDK target and mounts remote volume
- “*nqn*, *ip*, *port*, *diskid*, *iqn*” are passed in from Controller Driver

| CSI Message | Node (NVMf) | Node (iSCSI) |
|-----------------|--|---|
| StageVolume | nvme connect -n <i>nqn</i> -a <i>ip</i> -s <i>port</i> ... mount /dev/disk/by-id/ <i>diskid</i> stagePath | iscsiadm -p <i>ip:port</i> -m discovery ... iscsiadm -T <i>iqn</i> -p <i>ip:port</i> --login ... mount /dev/disk/by-id/ <i>diskid</i> stagePath |
| UnstageVolume | nvme disconnect -n <i>nqn</i> umount stagePath | iscsiadm -T <i>iqn</i> -p <i>ip:port</i> --logout ... umount stagePath |
| PublishVolume | mount -o bind stagePath podPath | mount -o bind stagePath podPath |
| UnpublishVolume | umount podPath | umount podPath |

Sequence Diagram





Community

Contributions Welcome

- Code review at SPDK Gerrit
 - *git clone <https://review.spdk.io/spdk/spdk-csi>*
 - Github mirror: <https://github.com/spdk/spdk-csi>
- Development Guidelines
 - <https://spdk.io/development/>
- Trello Board
 - <https://trello.com/b/nBuijzya/kubernetes-integration>
- Slack Channel
 - <https://spdk-team.slack.com/messages/containers>

References

- Container Storage Interface (CSI) Spec
 - <https://github.com/container-storage-interface/spec/>
- Kubernetes CSI Documentation
 - <https://kubernetes-csi.github.io/docs/>
- SPDK JSON-RPC
 - <https://spdk.io/doc/jsonrpc.html>
- SPDK-CSI Design Document
 - <https://tinyurl.com/spdkcsi-design-doc>



Thank you