DevUpps…die Platte ist voll

SVA

Wie man persistenten Storage anbindet

/ Whoami

TIM SCHÖLLHAMMER

Agile IT & Software Development

Tel: +49 6122 536-0
Fax: +49 6122 536-399
Mobil: +49 175 9375982

Email: tim.schoellhammer@sva.de

http://www.sva.de

Bolzstraße 7
70173 Stuttgart

CHRISTIAN FEY

Agile IT & Software Development

Tel: +49 6122 536-0
Fax: +49 6122 536-399
Mobil: +49 151 180 251 39

Email: christian.fey@sva.de

http://www.sva.de

Borsigstraße 26
65205 Wiesbaden

Persistente Daten und Volumes

SVA

/ Warum eigentlich?

Persistente Daten

- Können über verschiedene Wege abgebildet werden
  - Datenbank (SQL, noSQL, KV), Dateisystem, Objectstore,…
- Wer kümmert sich darum?
  - Datenbank „Abteilung“, Storage „Abteilung“
  - Schnittstelle?
- Persistente Daten in K8s Umgebung überhaupt notwendig?
  - Liegt doch alles im Git, oder? Bauen wir schnell neu auf!
  - etcd, images, logging, auditing
  - Was ist mit der Payload?
  - Wie hosten eigentlich die Datenbänker ihre Datenbanken? K8s?

Volumes

- Wie definiert sich ein Volume in K8s?
  - Ordner (mit oder ohne Inhalt)
  - Zugreifbar von Containern in einem Pod
  - Entstehung, Backend und Inhalt hängen vom Typ ab z.B.:
    - configMap
    - emptyDir
    - pvc
    - **Csi**
- **Wie sonst auch will ich einfach mein Deployment beschreiben und K8s stellt es her**
  - Nicht noch zwischendurch die „Abteilung“ Storage behelligen
  - Auch nicht NFS shares an irgendwelche K8s nodes mappen

CSI
Container Storage Interface

SVA

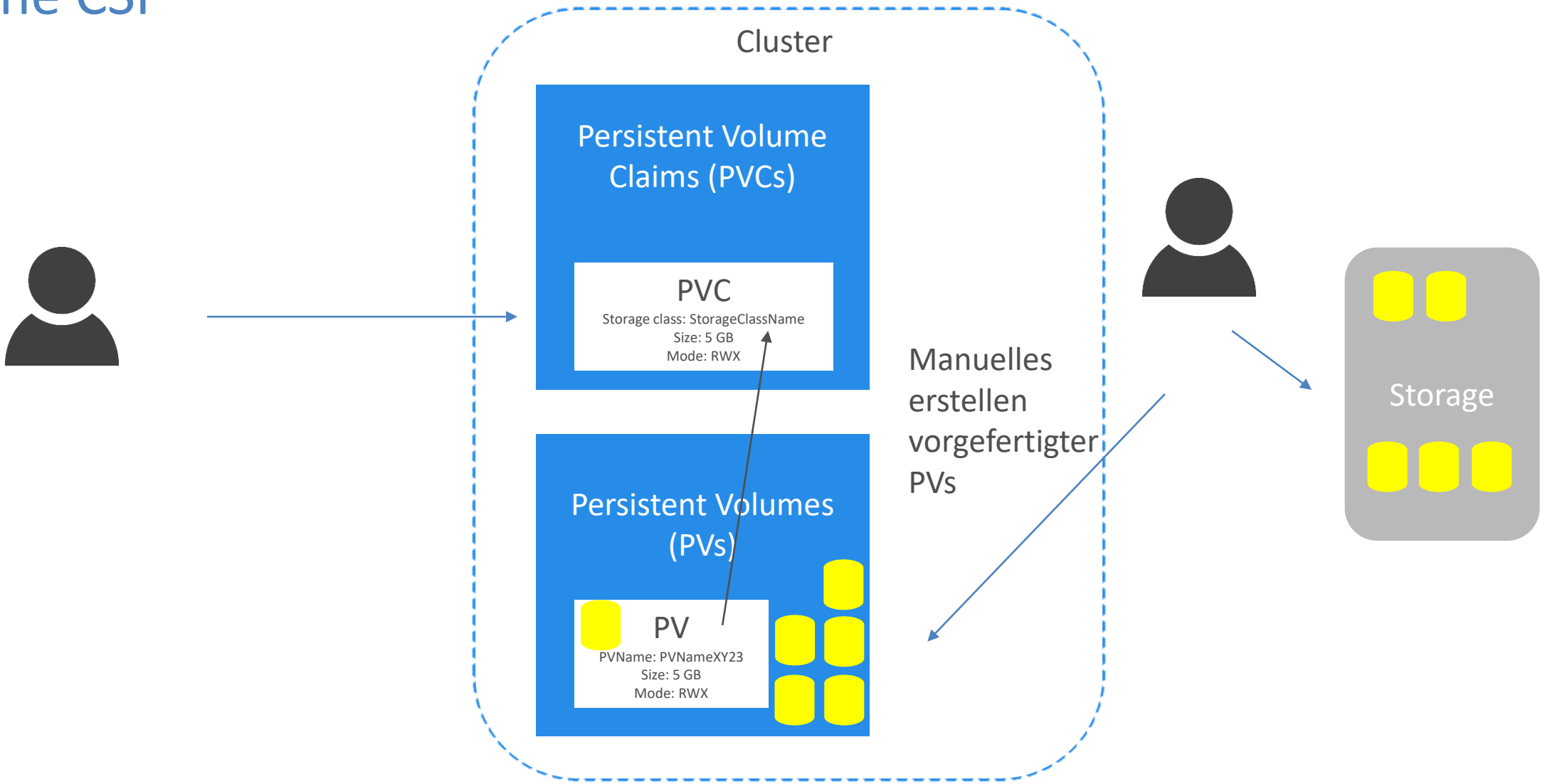
/ Warum geht es hier?

> *…define an industry standard “Container Storage Interface” (CSI) that will enable storage vendors (SP) to develop a plugin once and have it work across a number of container orchestration (CO) systems.*

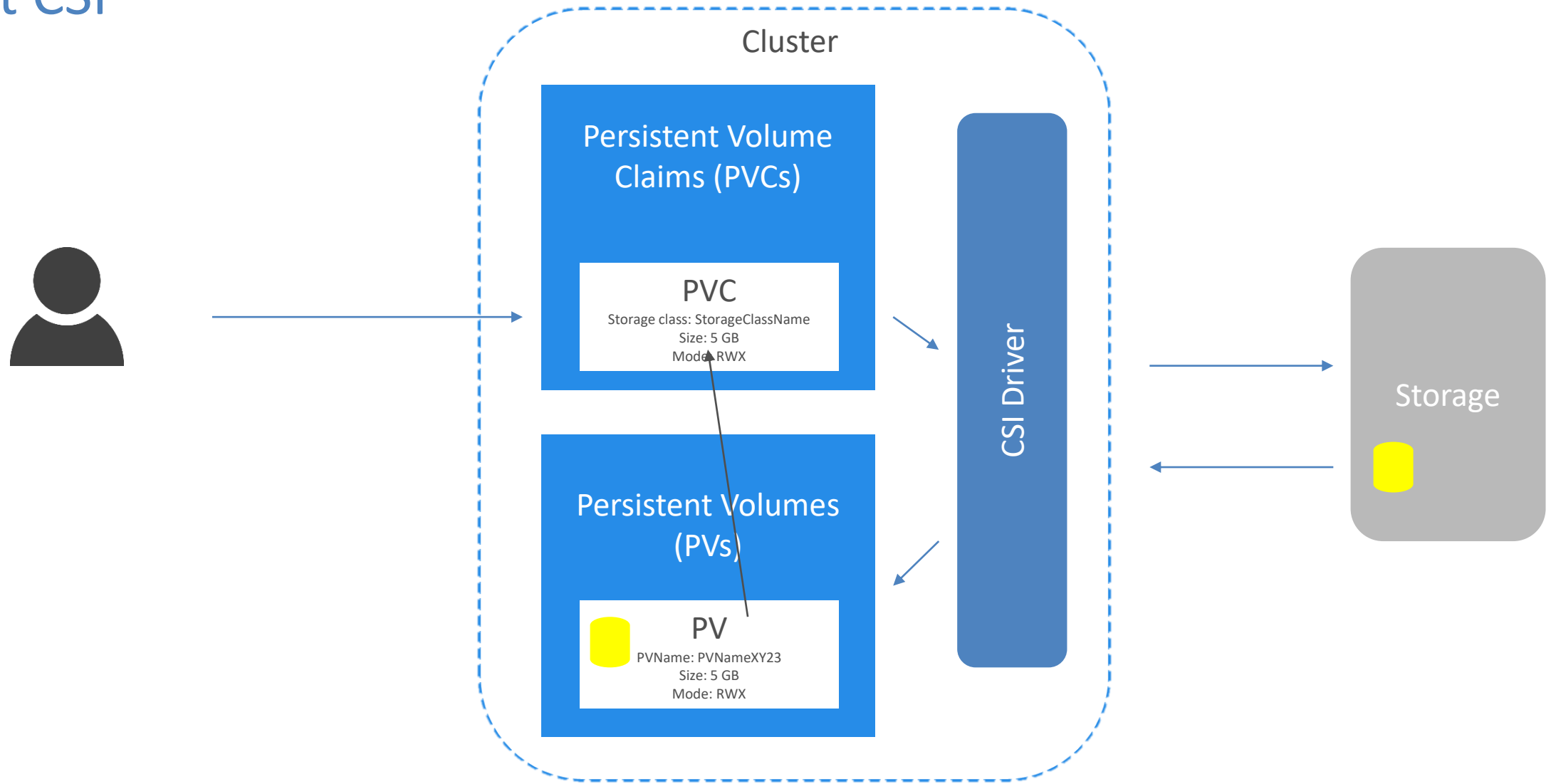
CSI in Kubernetes

- Es gab ja schon ein volume plugin system
  - In-tree -> Teil vom K8s Code -> release Zyklus
  - Schwer zu testen (zu wenig Platz unterm Schreibtisch)
- Vorteil der Nutzung von CSI für K8s
  - CSI integrieren => Schnittstelle zu 3rd party geschaffen
  - Hersteller hängen sich alle an die CSI Schnittstelle
  - GA seit v1.13
- CSI hat eigene spec
  - ggf. features in CSI spec aber nicht in K8s (z.B. Snapshots)

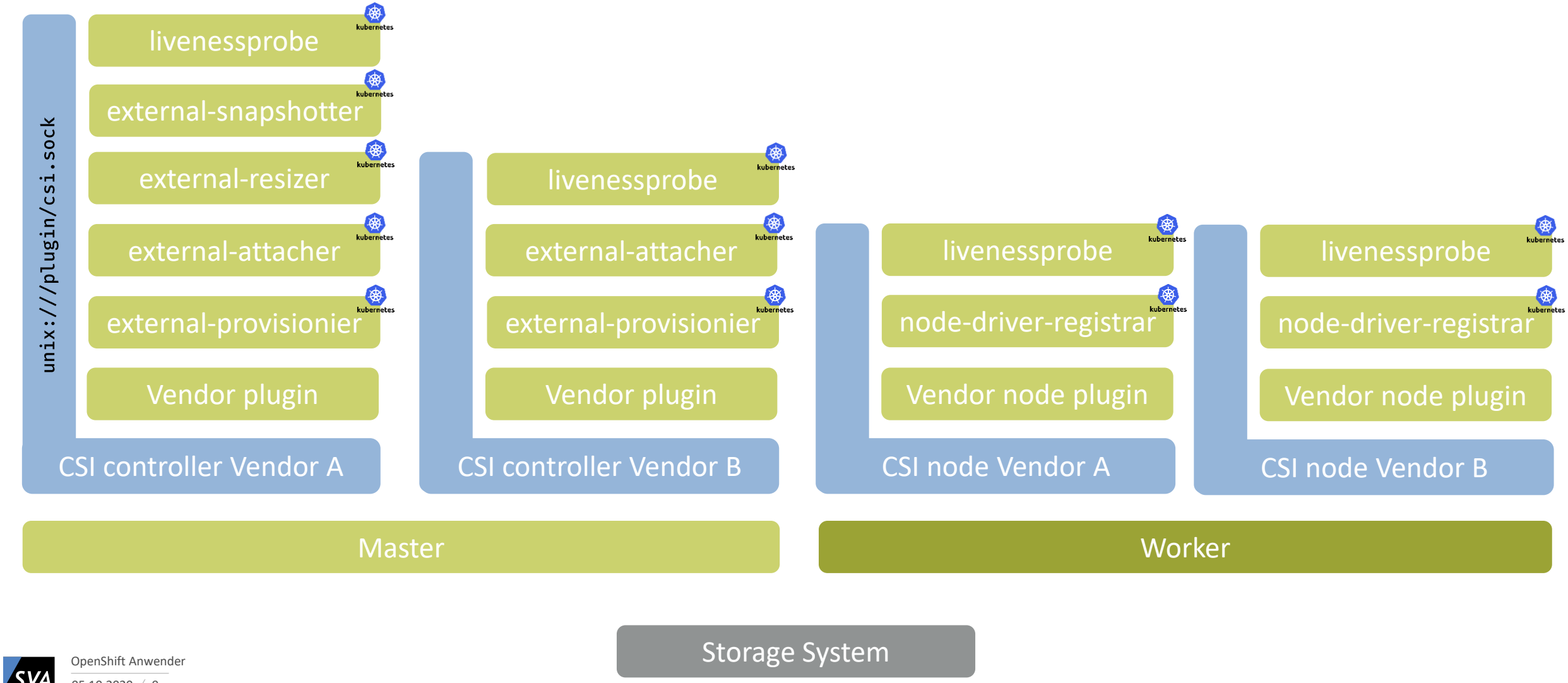
/ Ohne CSI


Cluster
Persistent Volume Claims (PVCs)
PVC
Storage class: StorageClassName
Size: 5 GB
Mode: RWX
Persistent Volumes (PVs)
PV
PVName: PVNameXY23
Size: 5 GB
Mode: RWX
Manuelles erstellen vorgefertigter PVs
Storage

/ Mit CSI

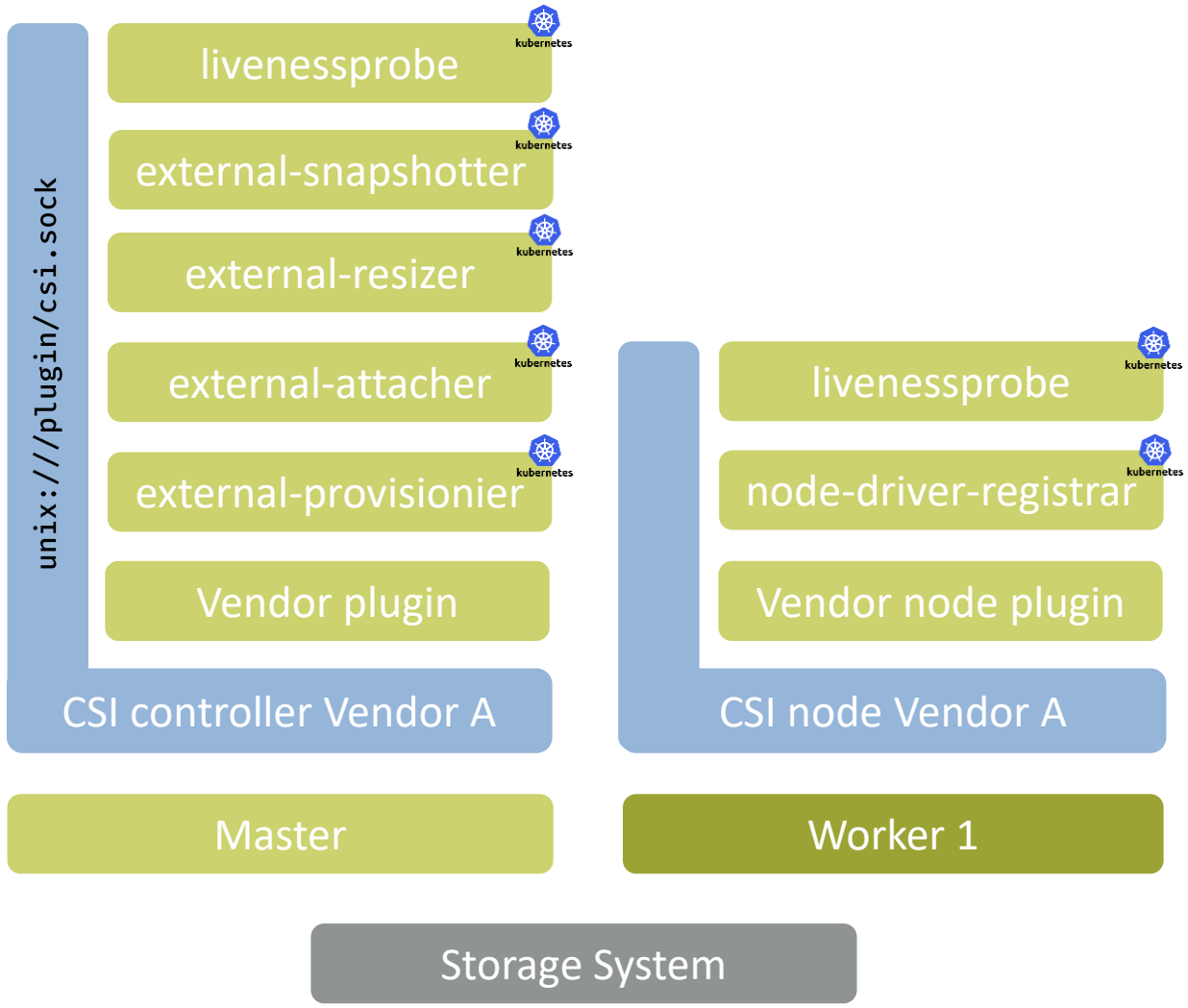

Cluster
Persistent Volume Claims (PVCs)
PVC
Storage class: StorageClassName
Size: 5 GB
Mode: RWX
Persistent Volumes (PVs)
PV
PVName: PVNameXY23
Size: 5 GB
Mode: RWX
CSI Driver
Storage

/ Topologie


unix:///plugin/csi.sock
livenessprobe
external-snapshotter
external-resizer
external-attacher
external-provisionier
Vendor plugin
CSI controller Vendor A
livenessprobe
external-attacher
external-provisionier
Vendor plugin
CSI controller Vendor B
livenessprobe
node-driver-registrar
Vendor node plugin
CSI node Vendor A
livenessprobe
node-driver-registrar
Vendor node plugin
CSI node Vendor B
Master
Worker
Storage System

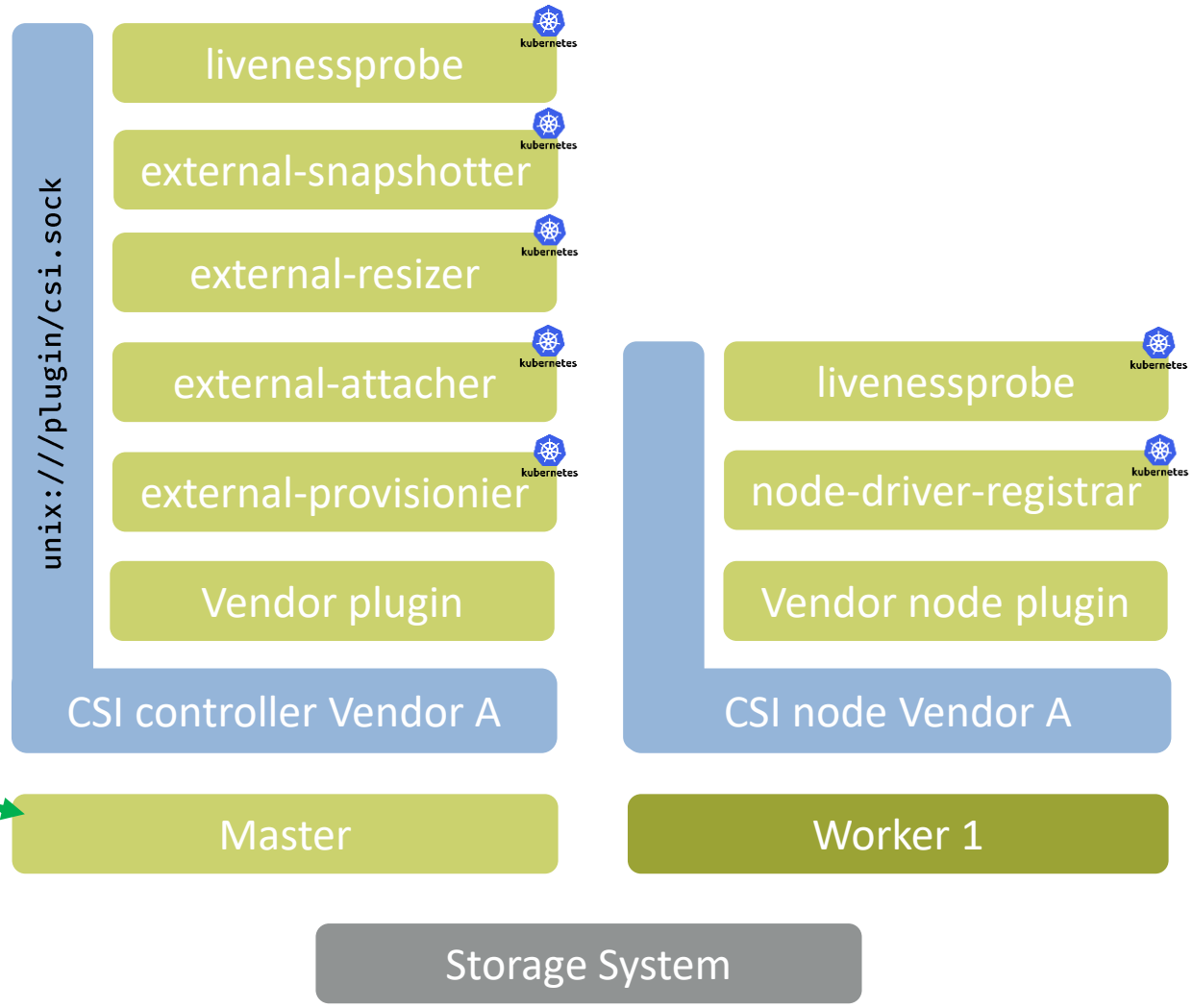
/ Volume Provisionierung

```
# cat pvc-basic.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my-first-volume
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: my-storageclass

# kubectl create -f pvc-basic.yaml
persistentvolumeclaim/affclaim created
```



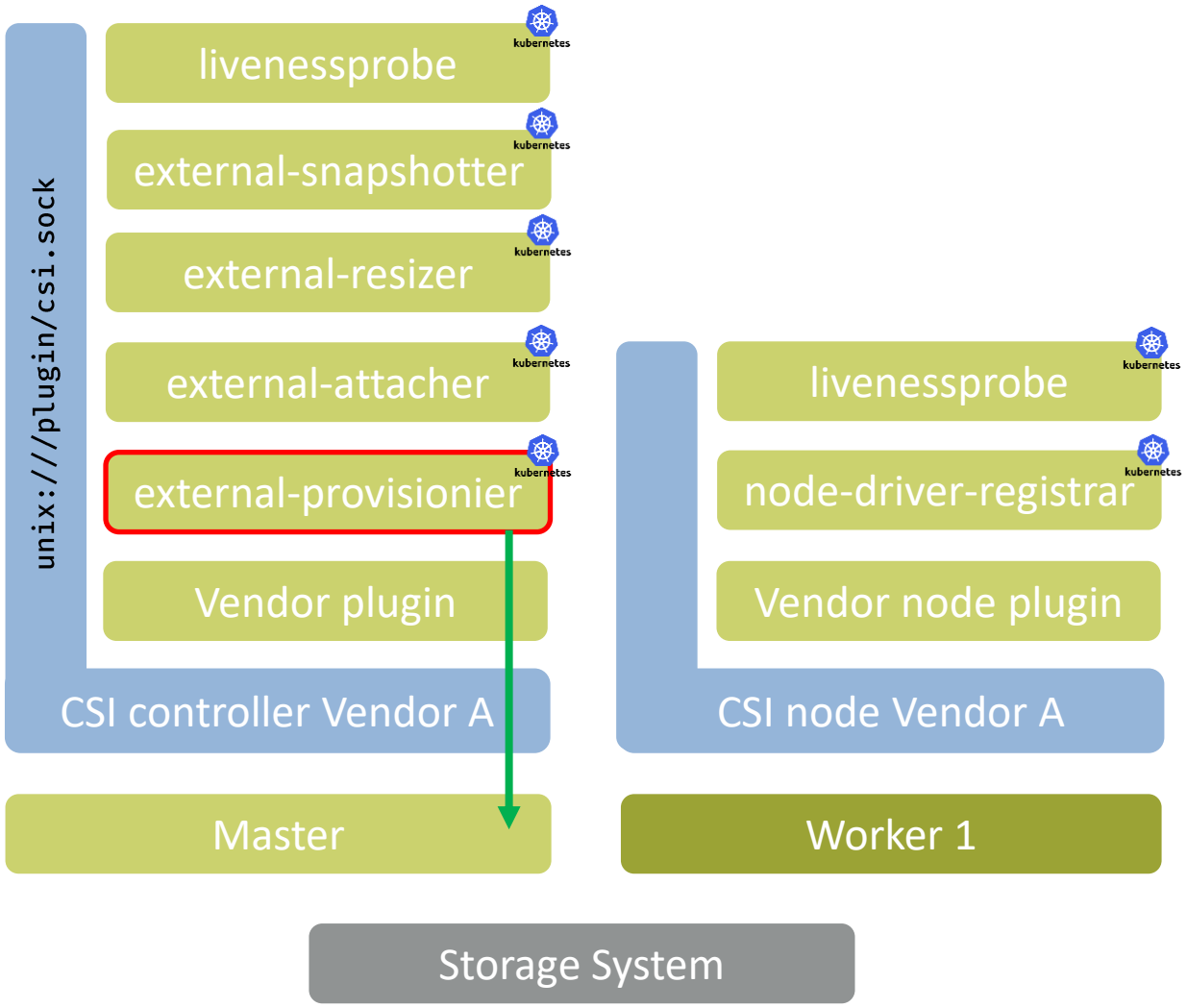
/ Volume Provisionierung

```
# cat pvc-basic.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my-first-volume
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: my-storageclass

# kubectl create -f pvc-basic.yaml
persistentvolumeclaim/affclaim created
```



/ Volume Provisionierung

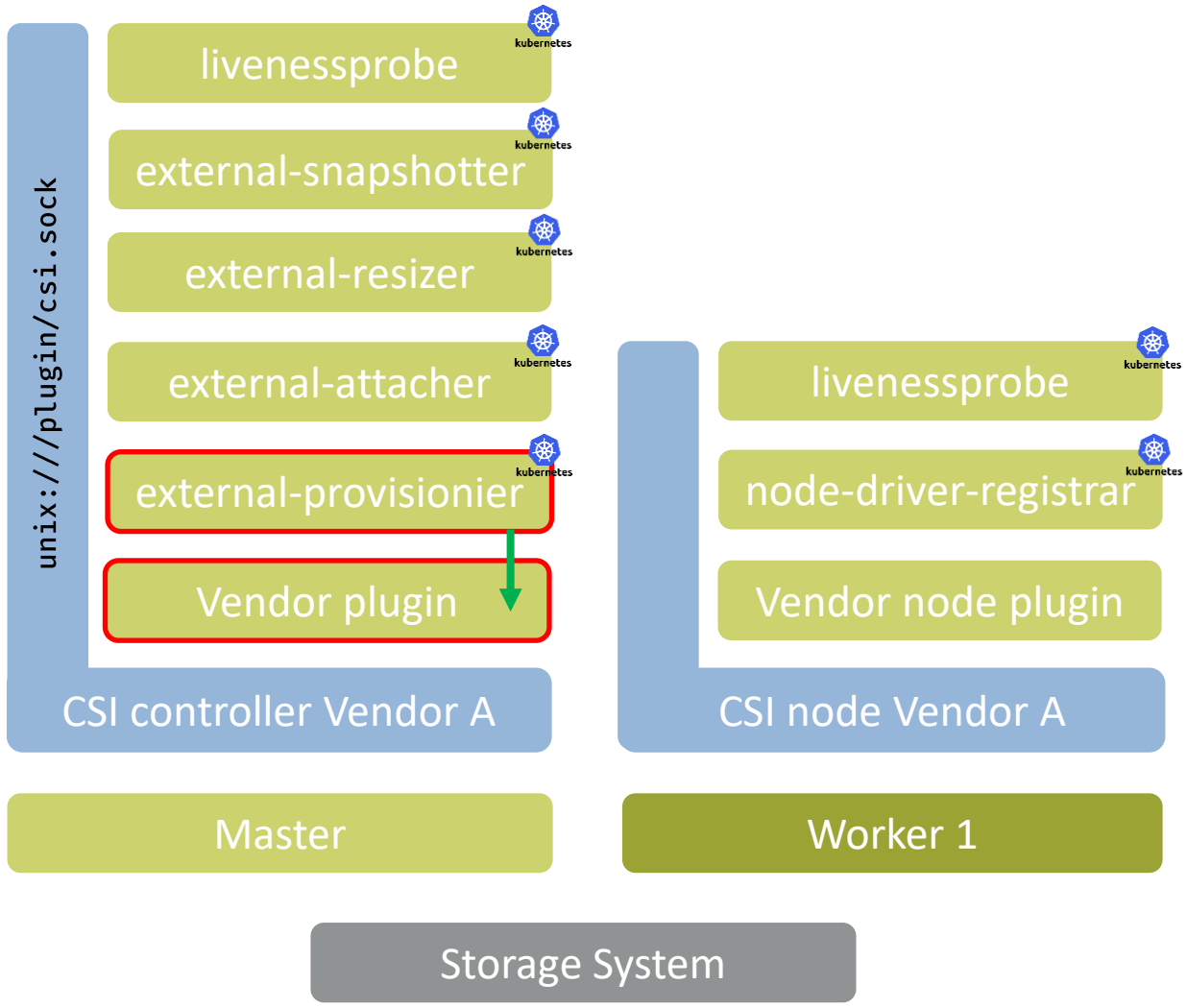
- Lauscht auf neues `PersistentVolumeClaim` Objekt



/ Volume Provisionierung

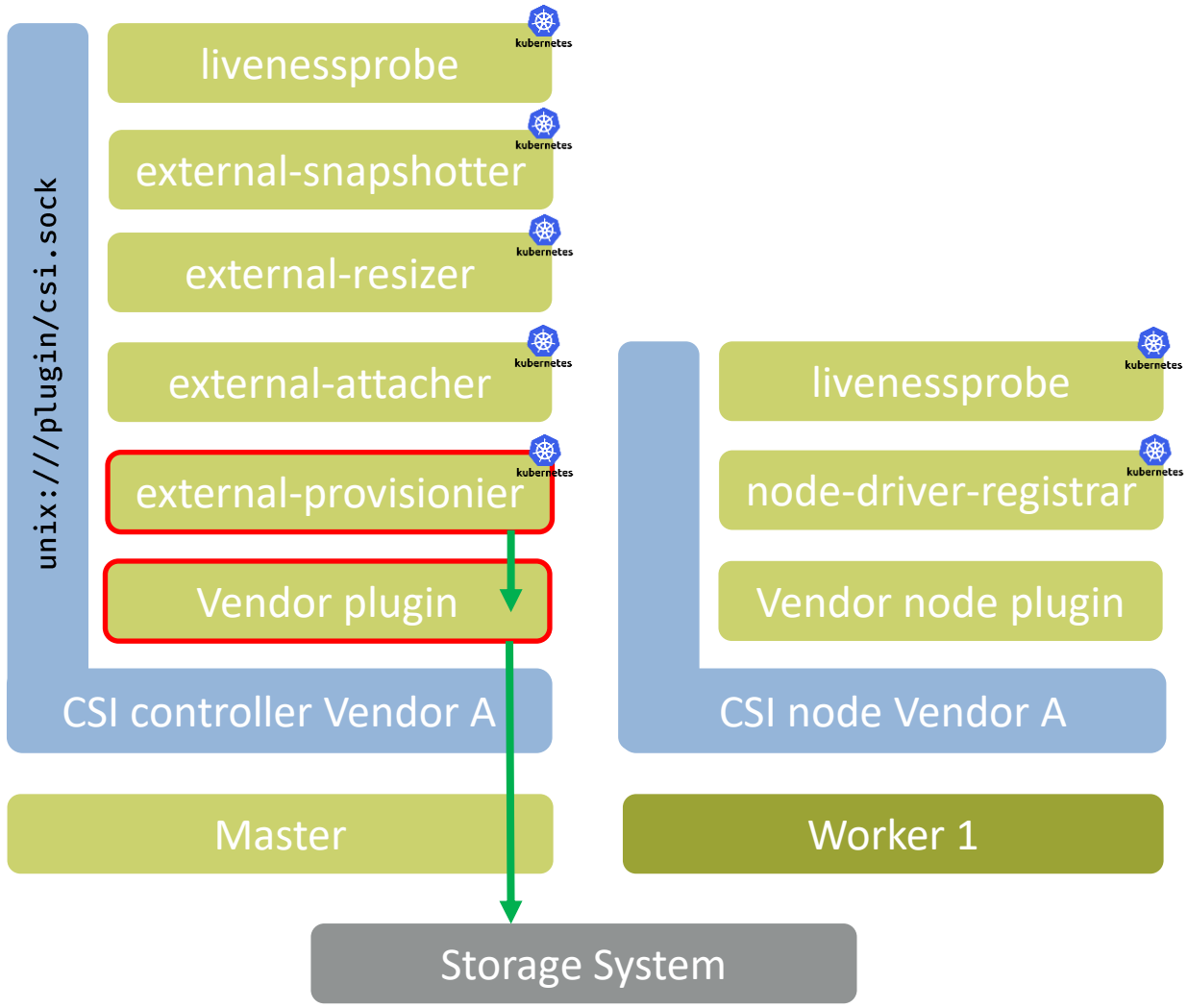
- Lauscht auf neues `PersistentVolumeClaim` Objekt

- Gibt den Request `CreateVolume` zu dem CSI Endpunkt



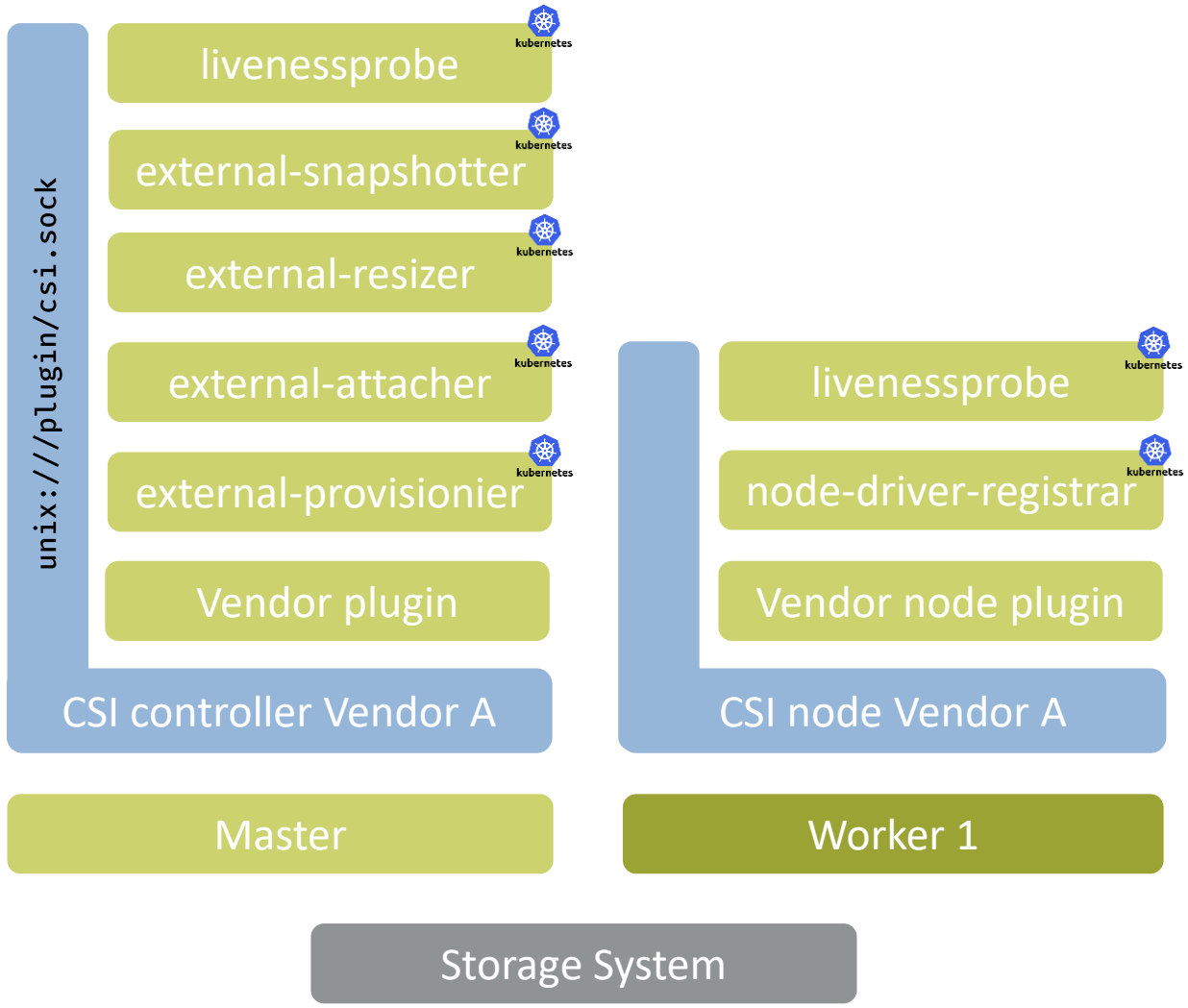
/ Volume Provisionierung

- Lauscht auf neues `PersistentVolumeClaim` Objekt
- Gibt den Request `CreateVolume` zu dem CSI Endpunkt
- Vendor plugin rennt los und macht die spezifische Arbeit auf dem Storage



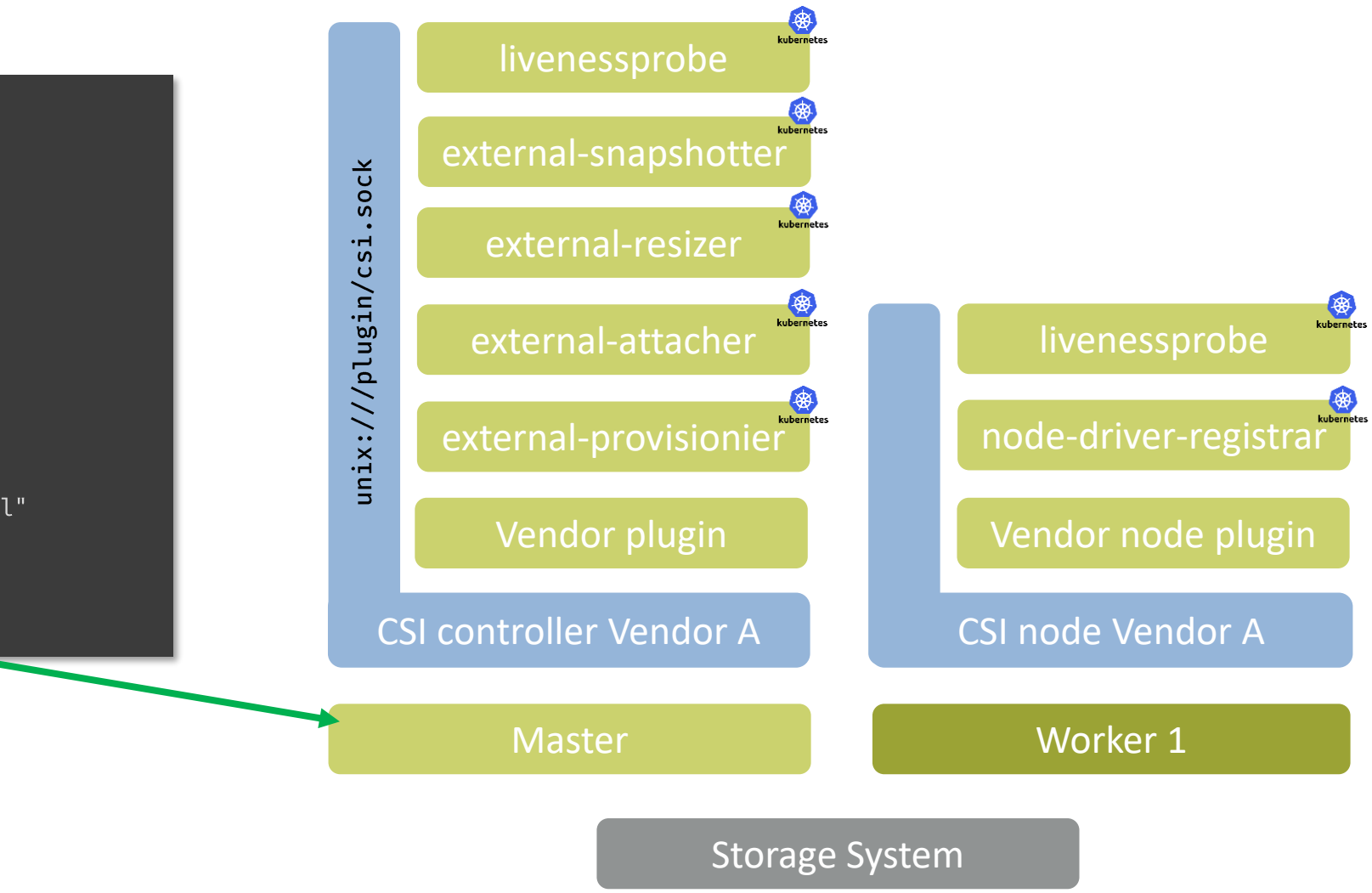
/ Volume Provisionierung

```
# cat nginx.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  volumes:
    - name: nginx-volume
      persistentVolumeClaim:
        claimName: my-first-volume
  containers:
    - image: nginx
  volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: nginx-volume

# kubectl create -f nginx.yaml
pod/nginx created
```



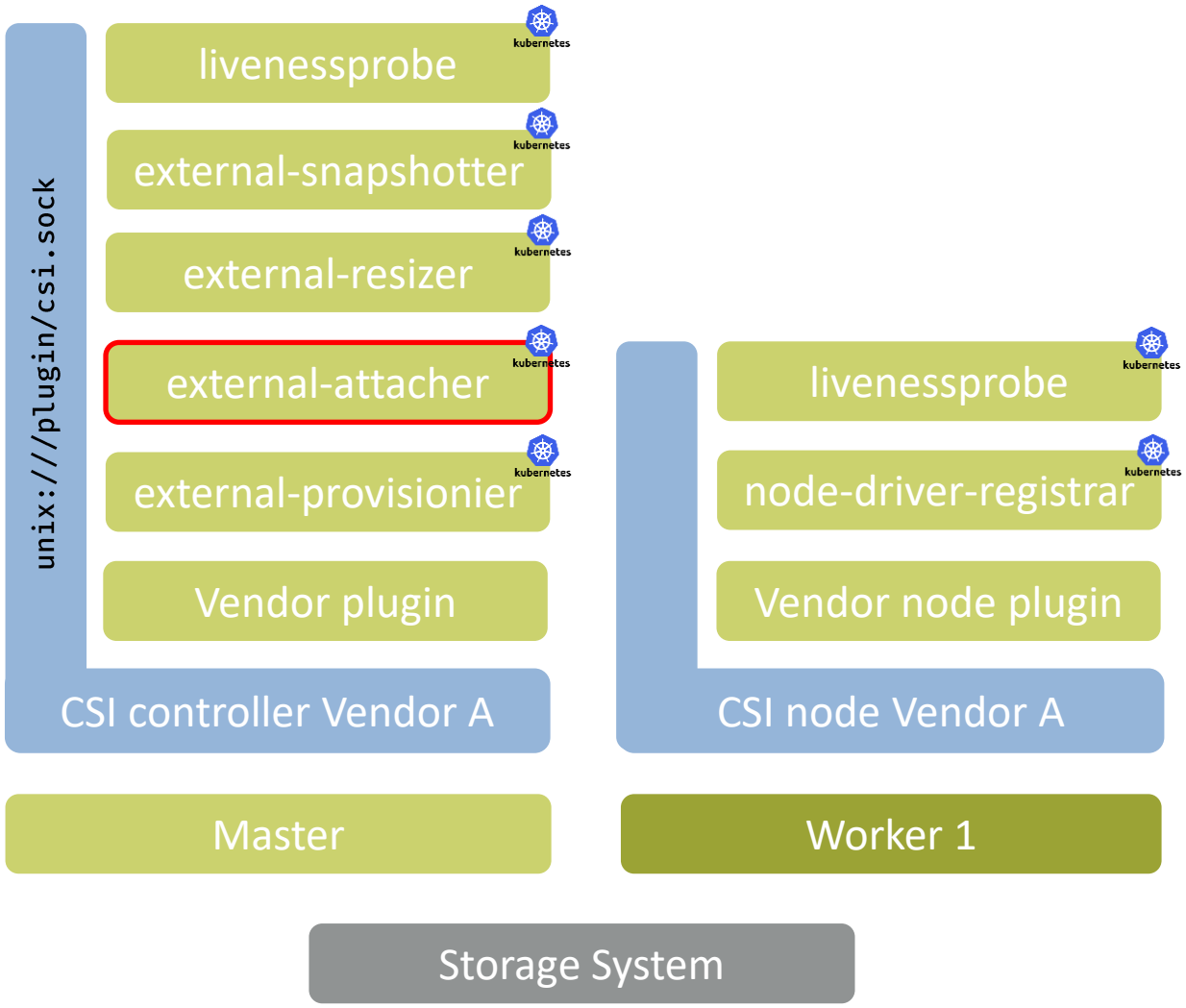
/ Volume Provisionierung

```
# cat nginx.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  volumes:
    - name: nginx-volume
      persistentVolumeClaim:
        claimName: my-first-volume
  containers:
    - image: nginx
  volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: nginx-volume

# kubectl create -f nginx.yaml
pod/nginx created
```



/ Volume Provisionierung

- Lauscht auf neues `VolumeAttachment` Objekt

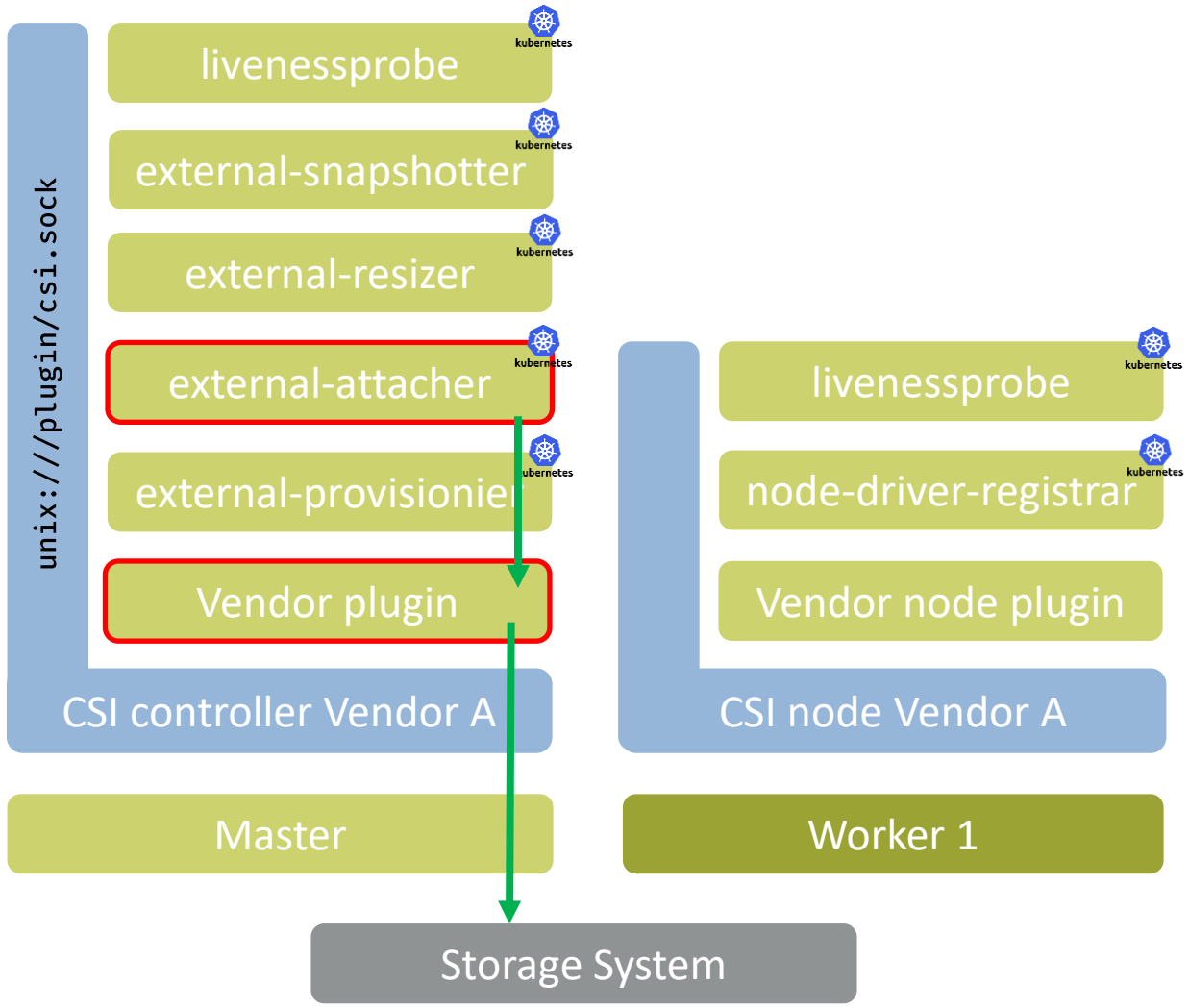


/ Volume Provisionierung

- Lauscht auf neues `VolumeAttachment` Objekt

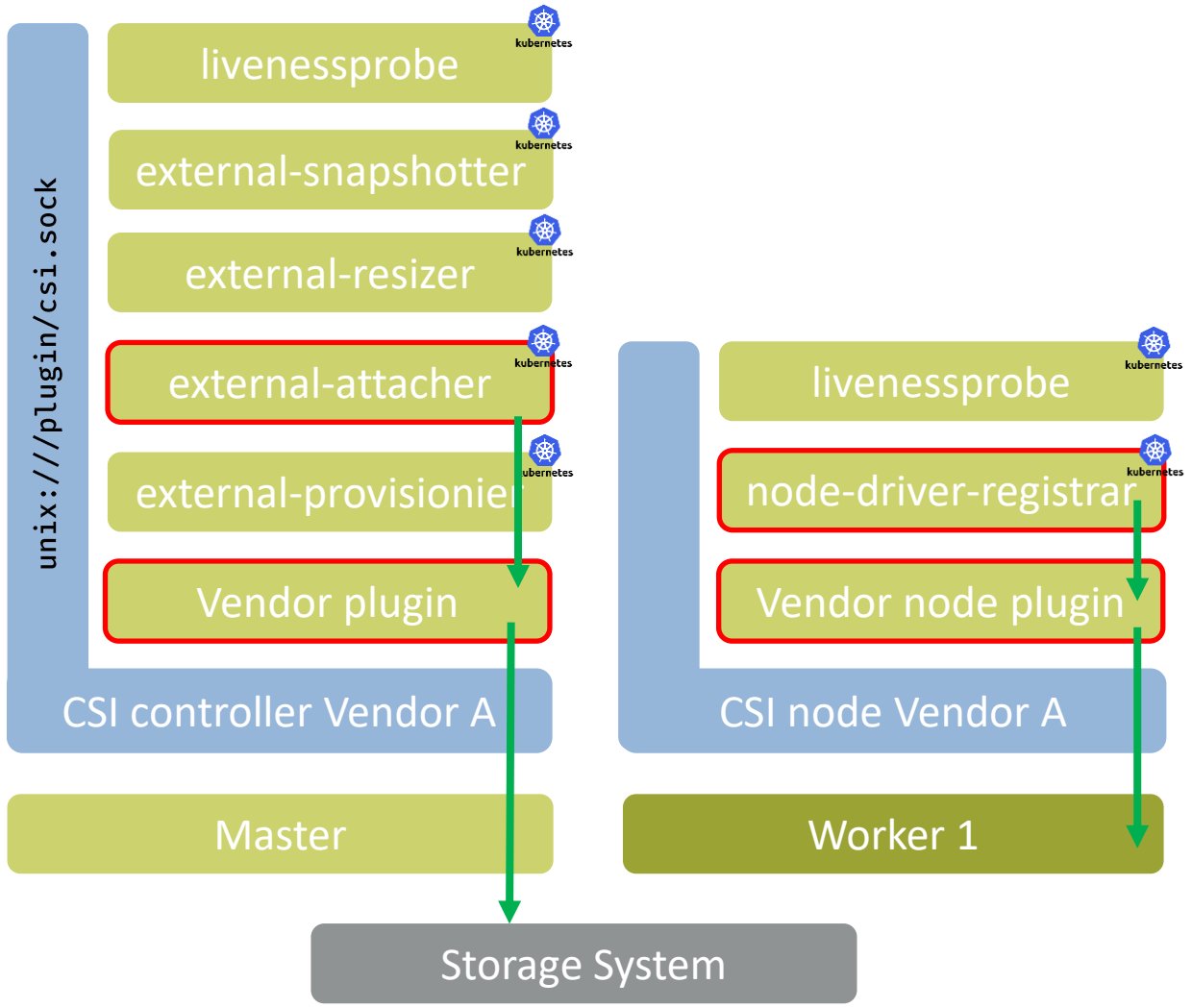
- Führt eine Controller Publish|Unpublish Operation gegen den CSI Endpunkt durch

- Storage Mapping / Attachment



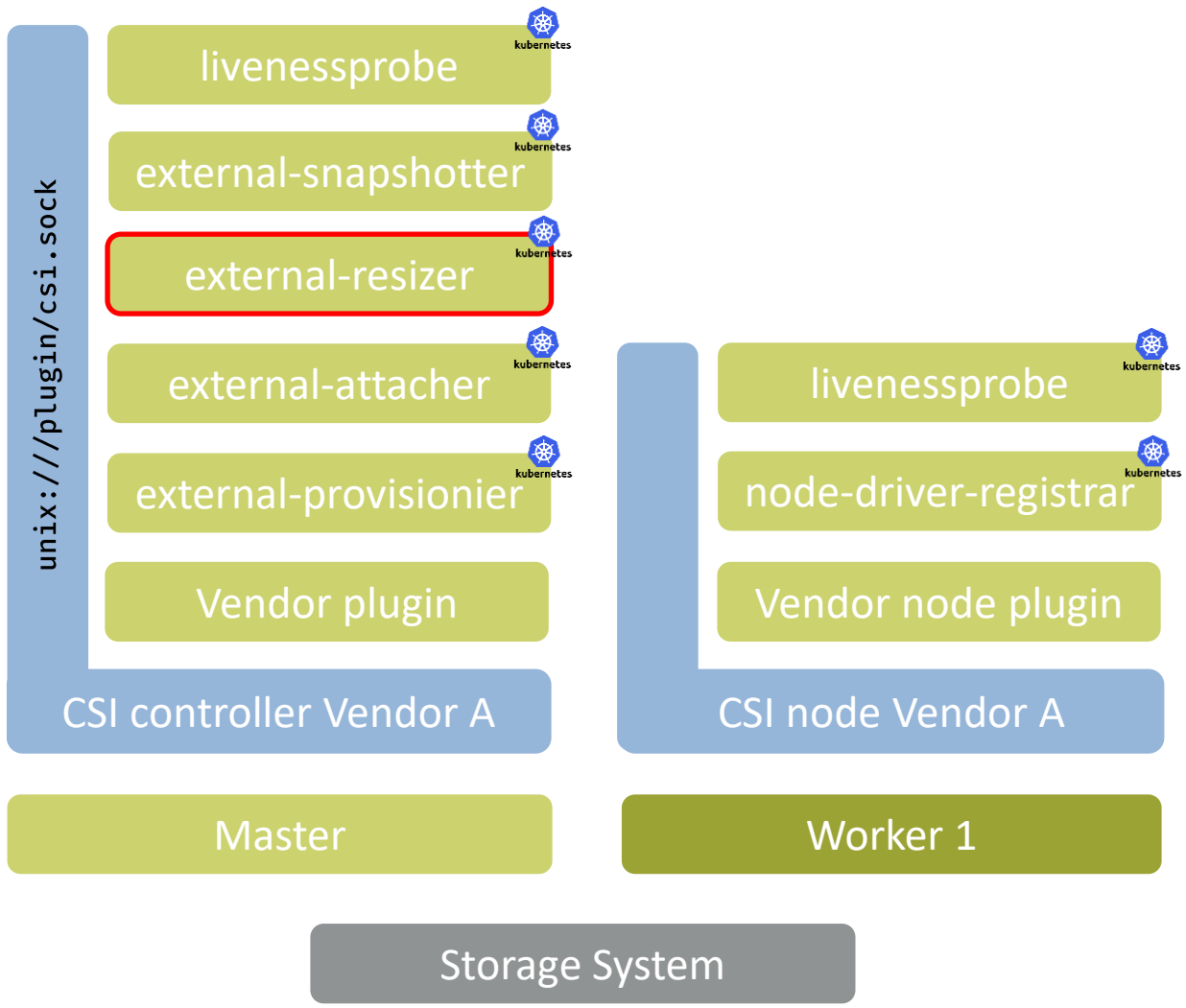
/ Volume Provisionierung

- Lauscht auf neues `VolumeAttachment` Objekt
- Führt eine Controller Publish|Unpublish operation gegen den CSI Endpunkt durch
- Storage Mapping / Attachment (Device dem Node von außen anhängen)
- Eigentliche Mount Operation (kubelet / csi node)



/ Volume Vergrößerung

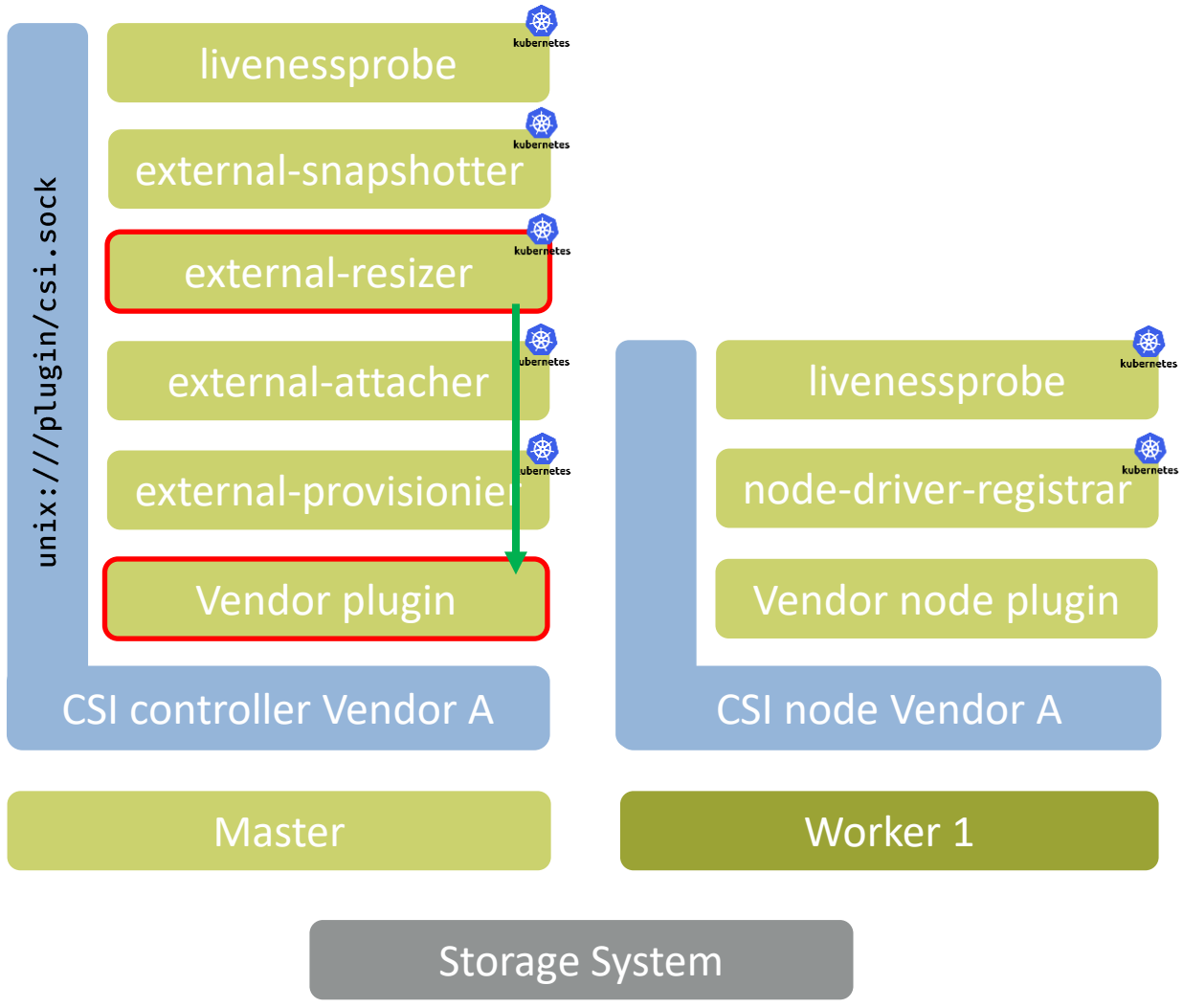
- Lauscht auf Updates am `PersistentVolumeClaim` Objekt



/ Volume Vergrößerung

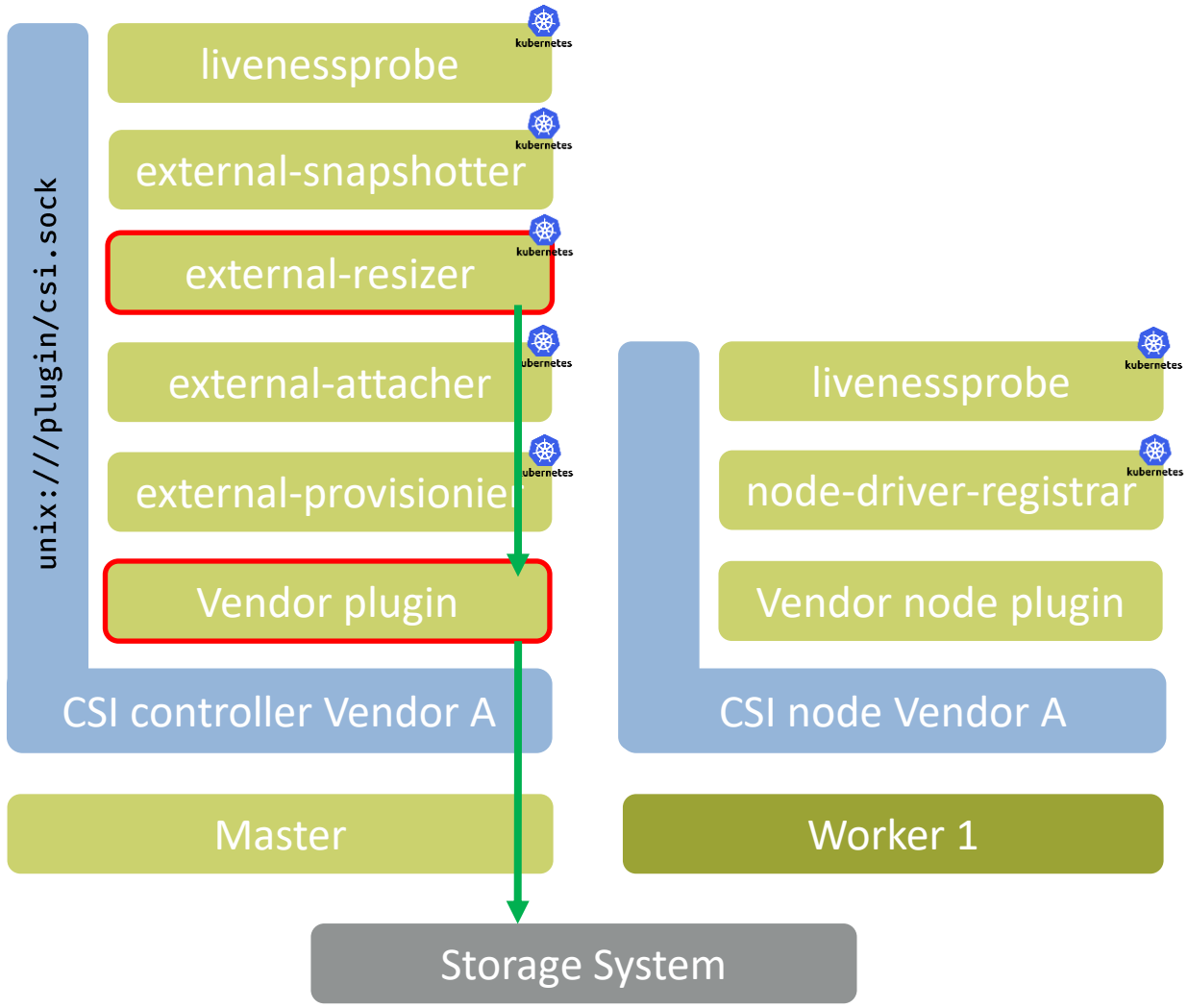
- Lauscht auf Updates am `PersistentVolumeClaim` Objekt

- Führt eine Controller `ExpandVolume` operation gegen den CSI Endpunkt durch



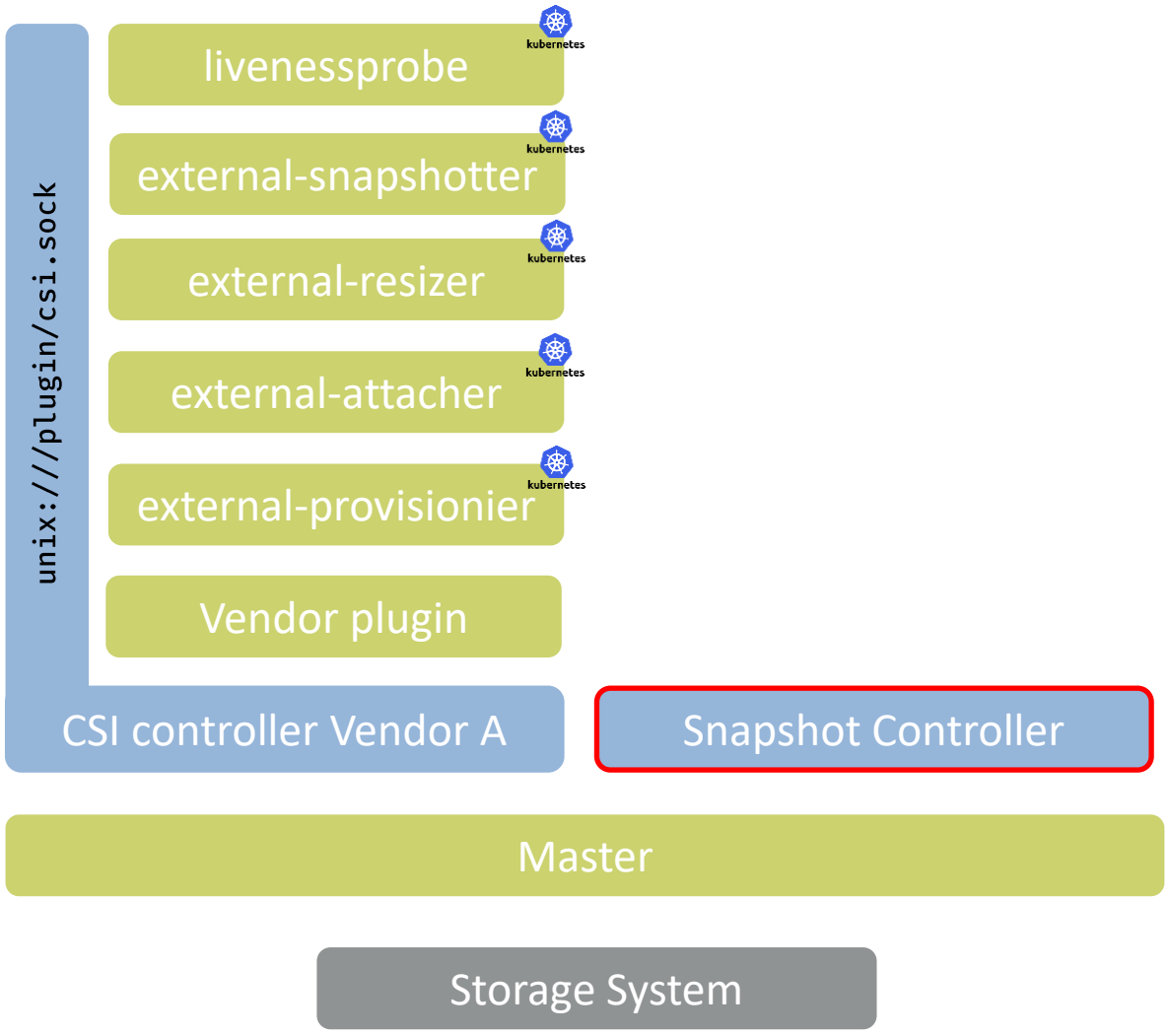
/ Volume Vergrößerung

- Lauscht auf Updates am `PersistentVolumeClaim` Objekt
- Führt eine Controller `ExpandVolume` operation gegen den CSI Endpunkt durch
- Dann Umsetzen der Konfiguration auf dem Storage System



/ Volume Snapshots

- Snapshot Controller lauscht auf `VolumeSnapshot` CRDs und erzeugt `VolumeSnapshotContent`

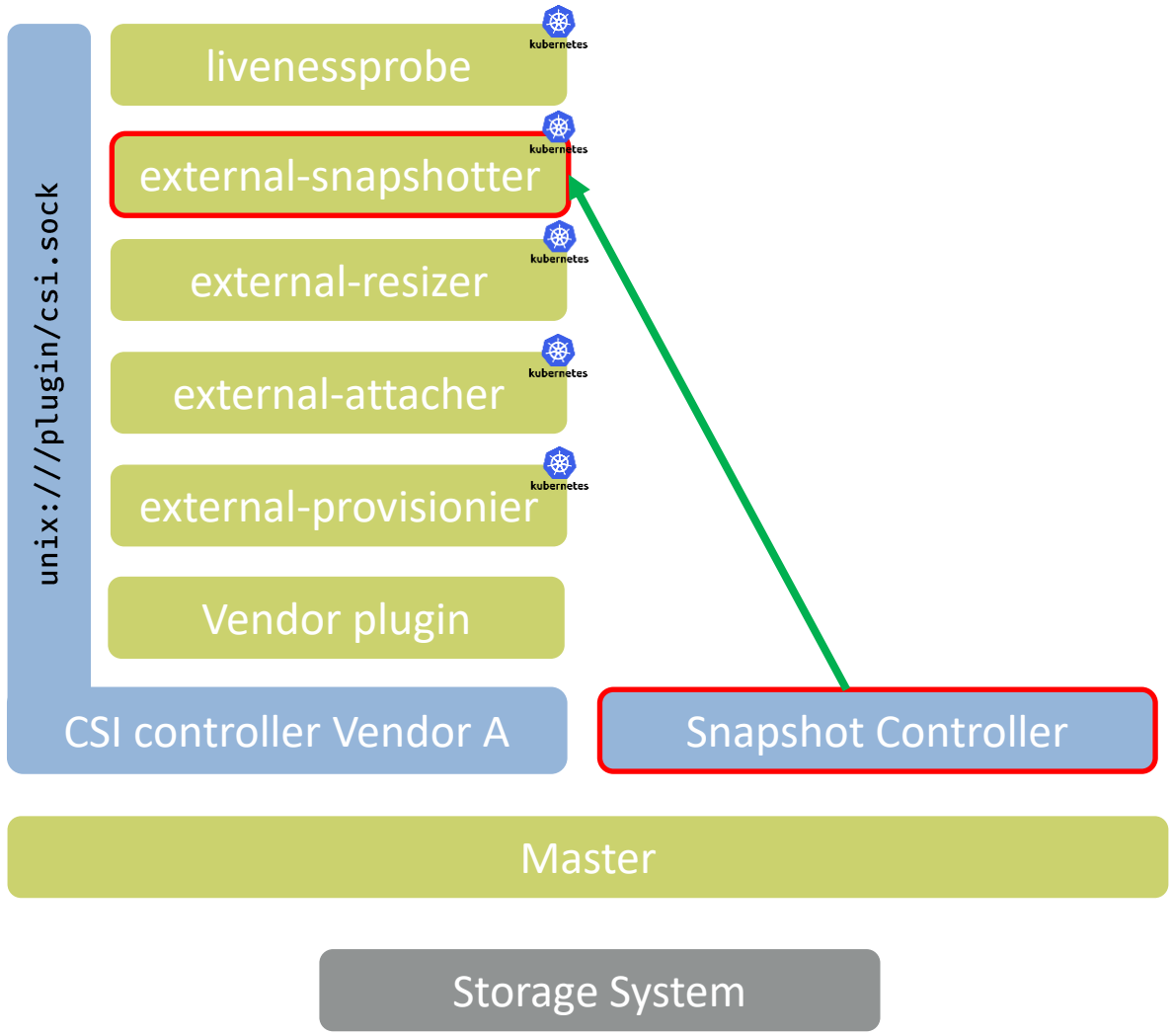


/ Volume Snapshots

- Snapshot Controller lauscht auf `VolumeSnapshot` CRDs und erzeugt `VolumeSnapshotContent`

- External snapshottter lauscht auf `VolumeSnapshotContent` CRDs

- Triggert dann den CSI controller



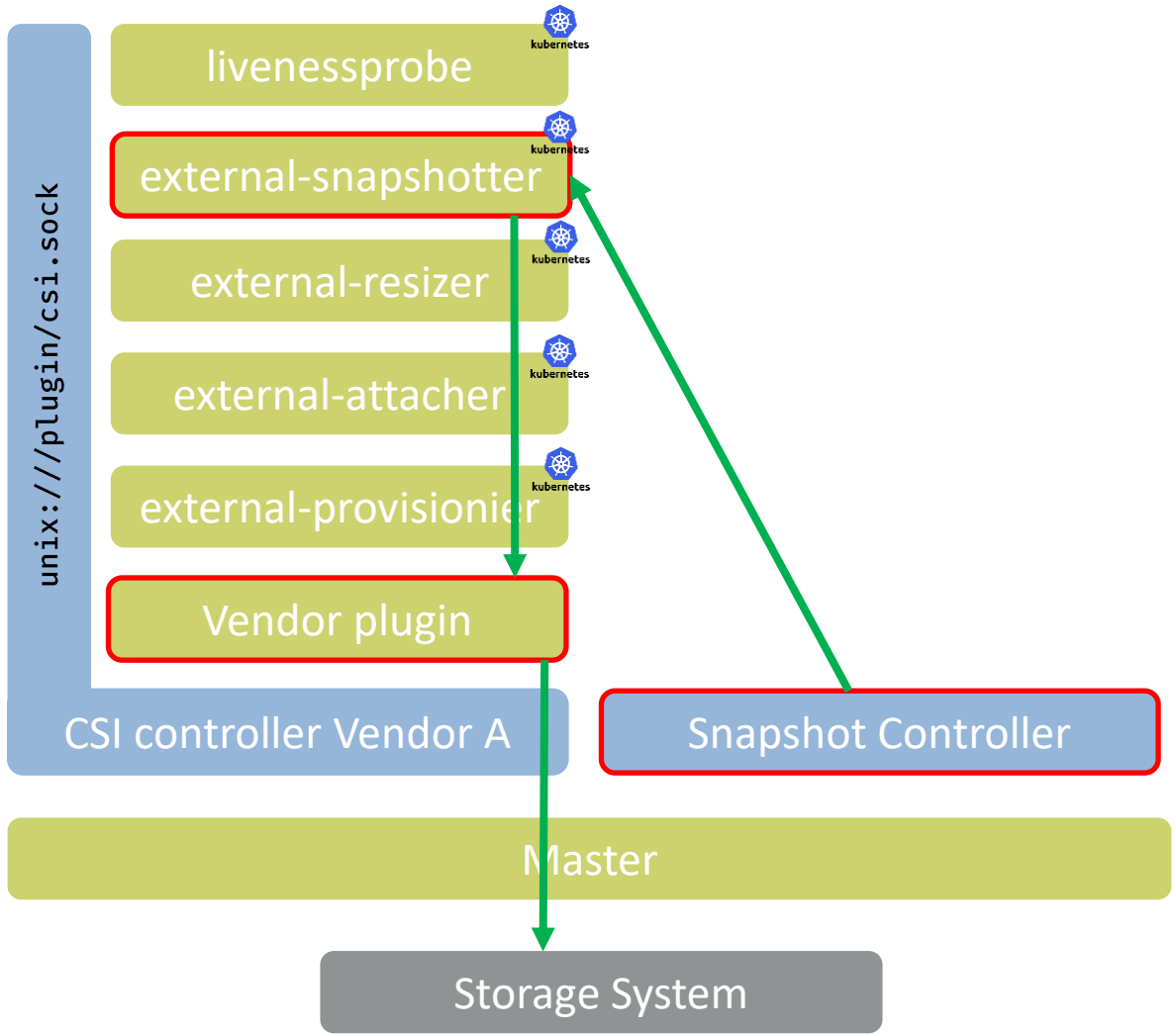
/ Volume Snapshots

- Snapshot Controller lauscht auf `VolumeSnapshot` CRDs und erzeugt `VolumeSnapshotContent`

- External snapshottter lauscht auf `VolumeSnapshotContent` CRDs

- Triggert dann den CSI controller

- Danach Umsetzung auf dem Storagesystem



Demo

SVA

Lessons learned aus dem Feld

/ RedHat CoreOS

Ist kein RHEL

- Finger weg vom Betriebssystem…
- Use machineconfig!

/ Netzwerk für Storage

Einfach ein Netzwerk für iSCSI

- Warum?
  - MTU 9000
  - Storage Netzwerk -> Routing ist 666
- Einfach NIC an RHCOS
  - VMs -> easy NIC an VM done?
  - Computer says NO!
- RHCOS way!
- Base64 4tw

```
BOOTPROTO=static
DEVICE=ens224
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
PROXY_METHOD=none
BROWSER_ONLY=no
DEFROUTE=no
IPV4_FAILURE_FATAL=no
IPV6INIT=no
NAME=ens224
MTU=9000
IPADDR=10.100.129.16
NETMASK=255.255.255.0
HWADDR=52:54:00:73:4f:00
```

```
{
  "apiVersion": "machineconfiguration.openshift.io/v1",
  "kind": "MachineConfig",
  "metadata": {
    "labels": {
      "machineconfiguration.openshift.io/role": "worker"
    },
    "name": "99-storage-network"
  },
  "spec": {
    "config":
    {
     "ignition": {
      "config": {},
      "timeouts": {},
      "version": "2.1.0"
     },
     "networkd": {},
     "passwd": {},
     "storage": {
      "files": [
        {
          "filesystem": "root",
          "path": "/etc/sysconfig/network-scripts/ifcfg-worker0-sn",
          "contents": {
            "source": "data:text/plain;charset=utf-8;base64,<IFCFG_WORKER0>",
            "verification": {}
          },
          "mode": 420
          },
        {
         "filesystem": "root",
         "path": "/etc/sysconfig/network-scripts/ifcfg-worker1-sn",
         "contents": {
          "source": "data:text/plain;charset=utf-8;base64,<IFCFG_WORKER1>",
          "verification": {}
         },
         "mode": 420
        },
        {
         "filesystem": "root",
         "path": "/etc/sysconfig/network-scripts/ifcfg-worker2-sn",
         "contents": {
          "source": "data:text/plain;charset=utf-8;base64,<IFCFG_WORKER2>",
          "verification": {}
         },
         "mode": 420
        }
       ]
      },
      "systemd": {}
    },
    "osImageURL": ""
  }
}
```

/ Konfigurier mal multipathd…

iSCSI Provider möchte mutlipathd

- Vorher -> yum install … RHCOS says no!
- Vertrauen in den MCO
- Base64 4tw
- systemd iscsid enable

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-ibm-attach
spec:
  config:
    ignition:
      version: 2.2.0
    storage:
      files:
        - path: /etc/multipath.conf
          mode: 384
          filesystem: root
          contents:
            source: data:,defaults%<GANZVIELTEXT>
            verification: {}
        - path: /etc/udev/rules.d/99-ibm-2145.rules
          mode: 420
          filesystem: root
          contents:
            source: data:,%23<GANZVIELTEXT>
            verification: {}
    systemd:
      units:
      - name: multipathd.service
        enabled: true
      # Uncomment the following lines if this MachineConfig will be used with iSCSI connectivity
      - name: iscsid.service
        enabled: true
```

/ Generell

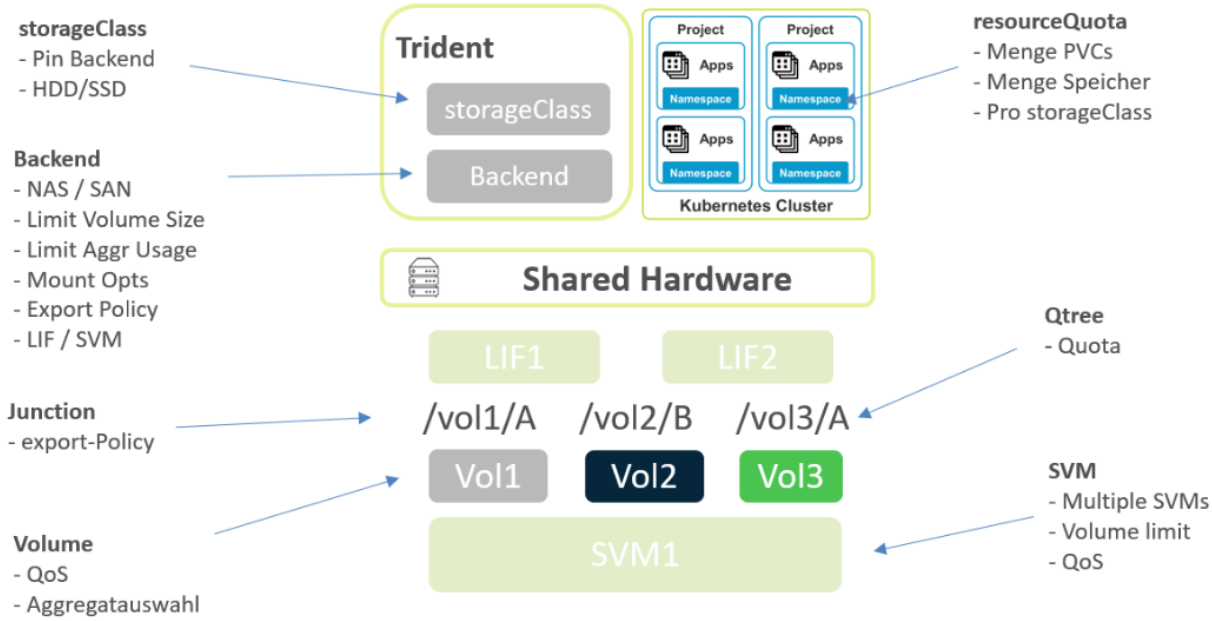
Kubernetes Namespaces

- PVC lebt im Namespace
  - PVC Namespace A
  - Clone Namespace A
  - PVC Namespace B based on Clone Namespace A…no -> Usecase?
- Authentication
  - iSCSI CHAP -> Yes
  - Kerberized NFS -> No(t yet) -> Sinnvoll?

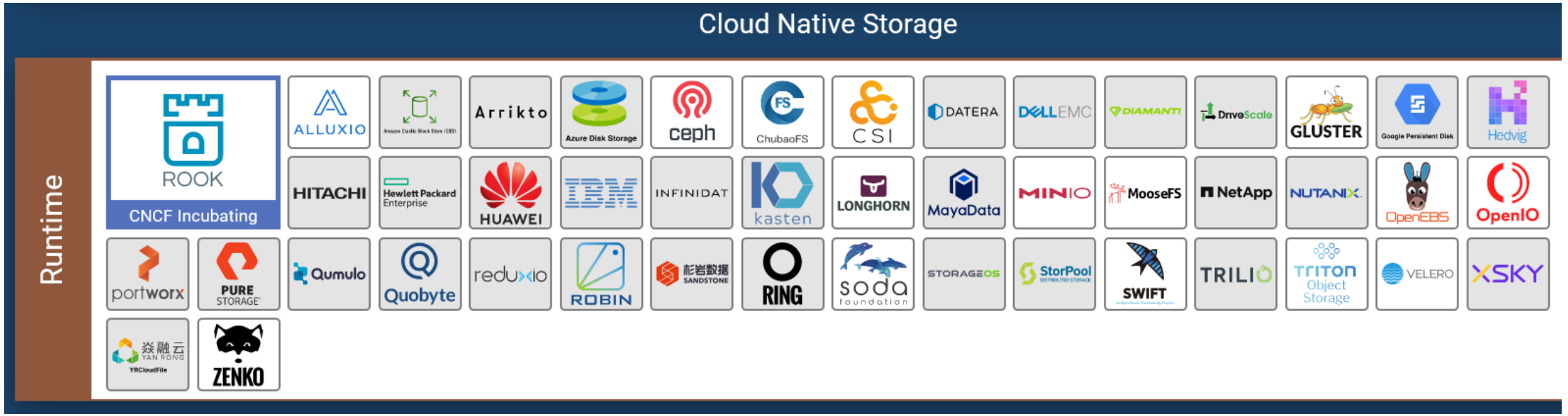
/ Limits & Safeguards

Ask your Storageteam

- Viele Storagesysteme bieten Multitenancy
  - Childpool (IBM)
  - Storage Virtual Machine (NetApp)
  - …
- Limits und Quotas
  - Quality of Service
  - Hard limits auf Storage System vs. „Soft-Limits“ in Plugin
- RBAC Roles
- RessourceQuota
  - StorageClass kann nicht „versteckt“ werden



/ Ecosystem


Cloud Native Storage
Runtime
ROOK
CNCF Incubating
ALLUXIO
Arrikto
Azure Disk Storage
ceph
ChubaoFS
CSI
DATERA
DELL EMC
DIAMANTI
DriveScale
GLUSTER
Google Persistent Disk
Hedvig
HITACHI
Hewlett Packard Enterprise
HUAWEI
IBM
INFINIDAT
kasten
LONGHORN
MayaData
MINIO
MooseFS
NetApp
NUTANIX
OpenEBS
OpenIO
portworx
PURE STORAGE
Qumulo
Quobyte
reduxio
ROBIN
彬岩数据 SANDSTONE
RING
soda foundation
STORAGEOS
StorPool
SWIFT
TRILIO
TRITON Object Storage
VELERO
XSKY
焱融云 YAN RONG
YRCloudFile
ZENKO

/ Features

common Persistent Volume modes, types, & capabilities

basic CSI driver Persistent Volume capabilities