


# Lessons Learned from Migrating Kubernetes from **Docker** to **containerd** runtime

PAYBASE\_



**Are You  
Missing out on  
the Fear of  
Missing Out?**



# whoami

Ana Calin

Systems Engineer @Paybase

Twitter: @AnaMariaCalin

LinkedIn: <https://www.linkedin.com/in/ana-maria-calin/>



# Table of Contents

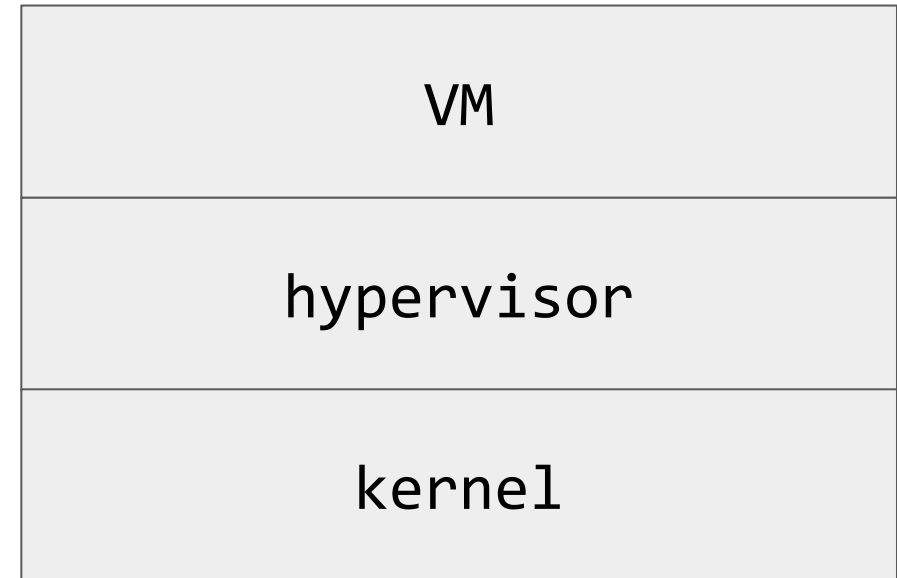
- 01 ~~whoami~~
- 02 What is a container runtime
- 03 Types of container runtimes
- 04 Docker, containerd and CRI-O comparison
- 05 Architecture overview
- 06 Migration Overview
- 07 User observations
- 09 Summary

# Understanding what a container runtime is

# What is a container runtime?

*A container runtime* is responsible for all the parts of running a container that isn't actually running the program itself.

# How we tend to think of containers

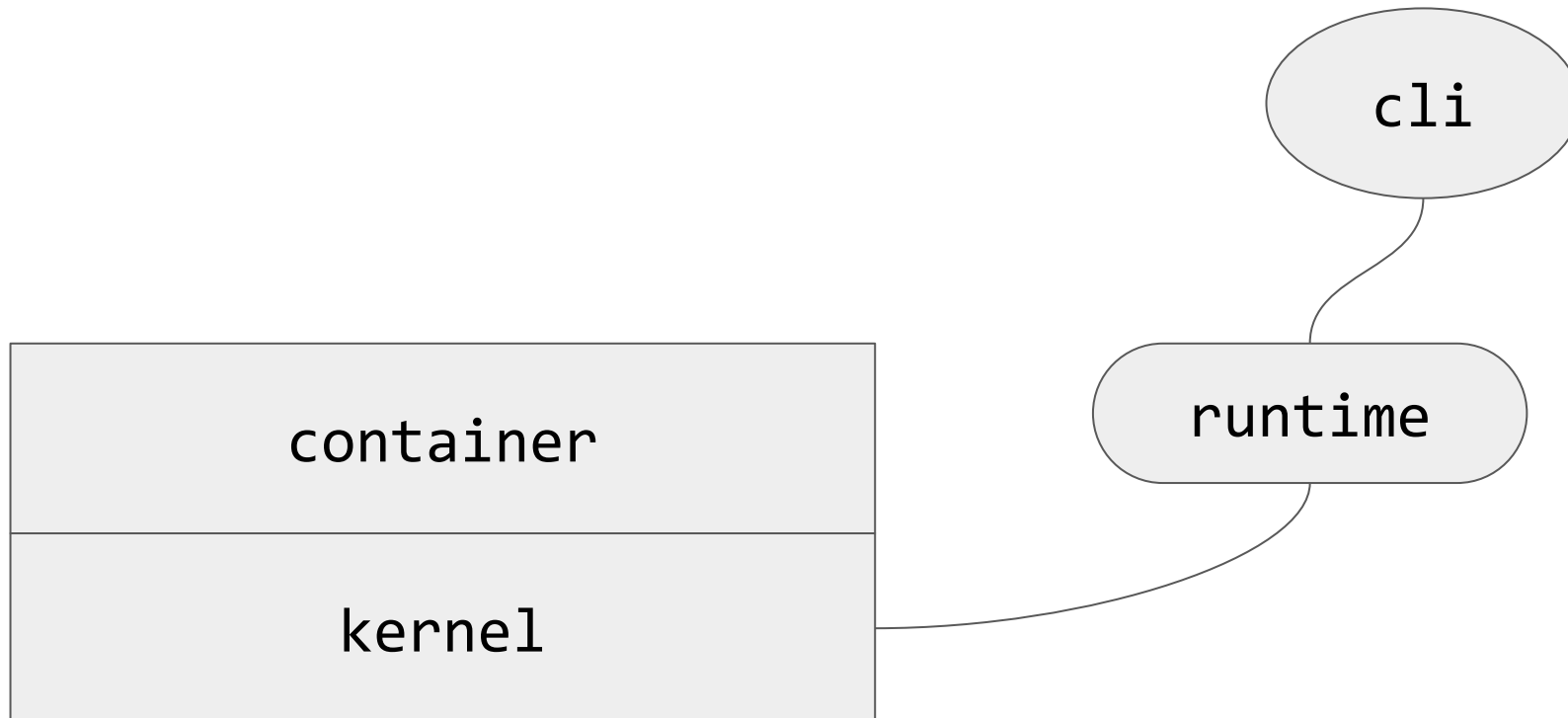


# How we tend to think of containers



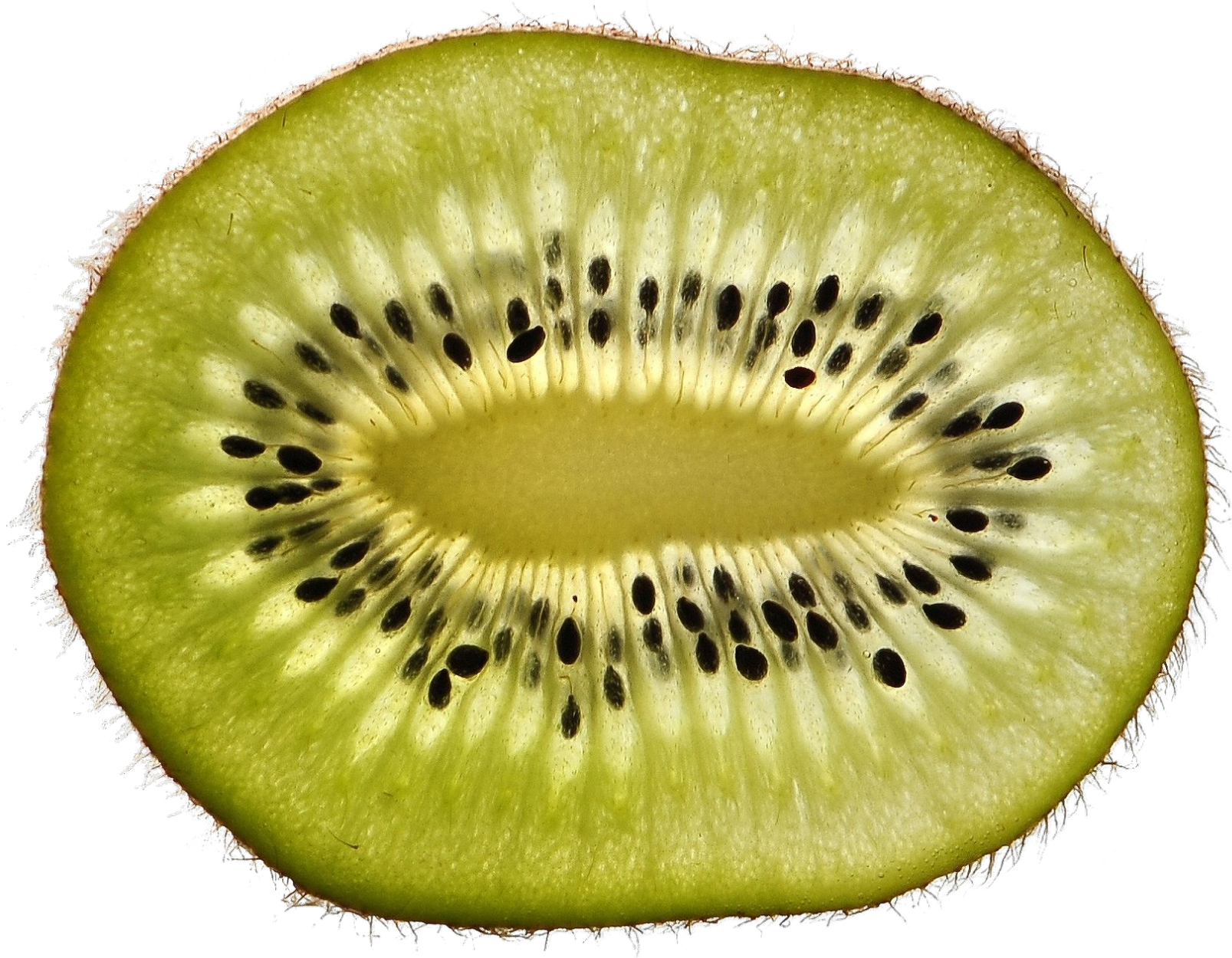


# More realistically



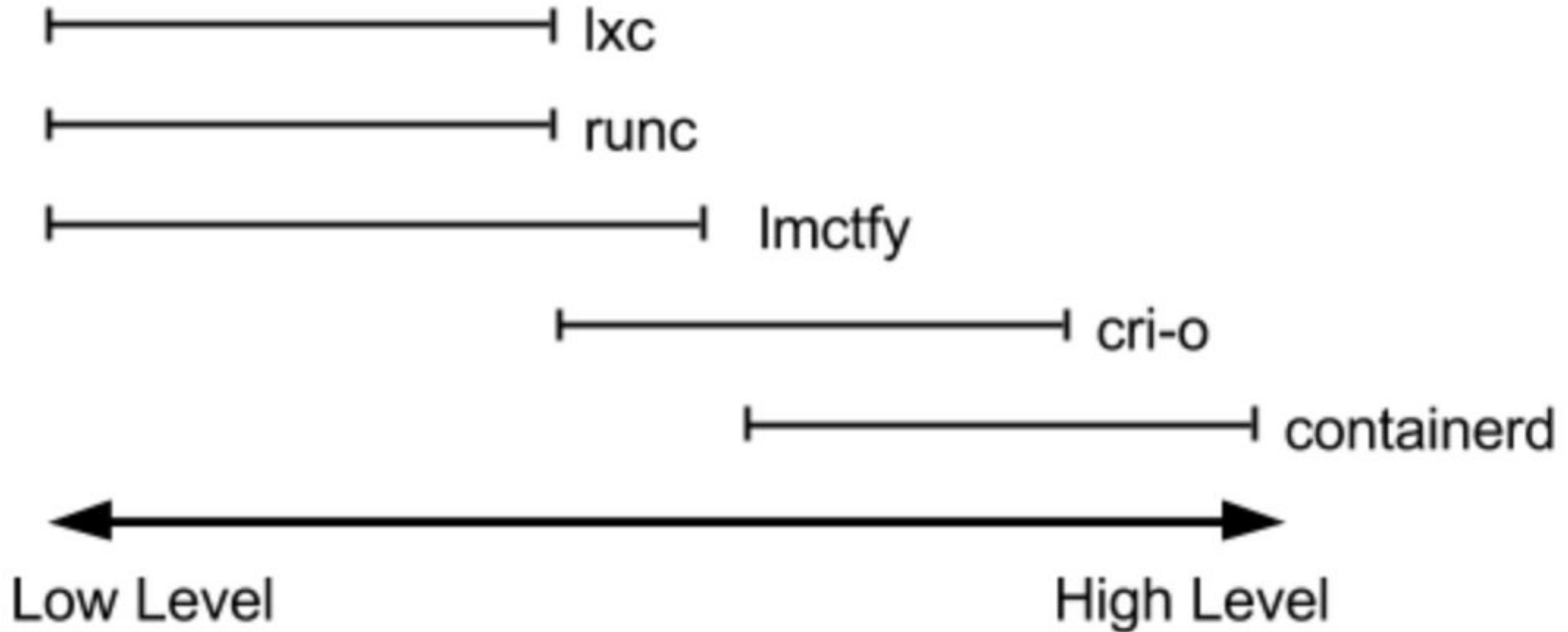
# Process Isolation

- **namespaces** - partitions kernel resources
  - *pid*
  - *net*
  - *mnt*
- **cgroups** - resource limits
- **seccomp-bpf**



# Types of container runtimes

# Low level vs High level



# What is a container runtime?

*A container runtime* is responsible for setting up namespaces and cgroups and then running commands inside those namespaces.

# More terminology

**CRI (Container Runtime Interface)**- a plugin interface which enables kubelet to use a wide variety of container runtimes, without the need to recompile

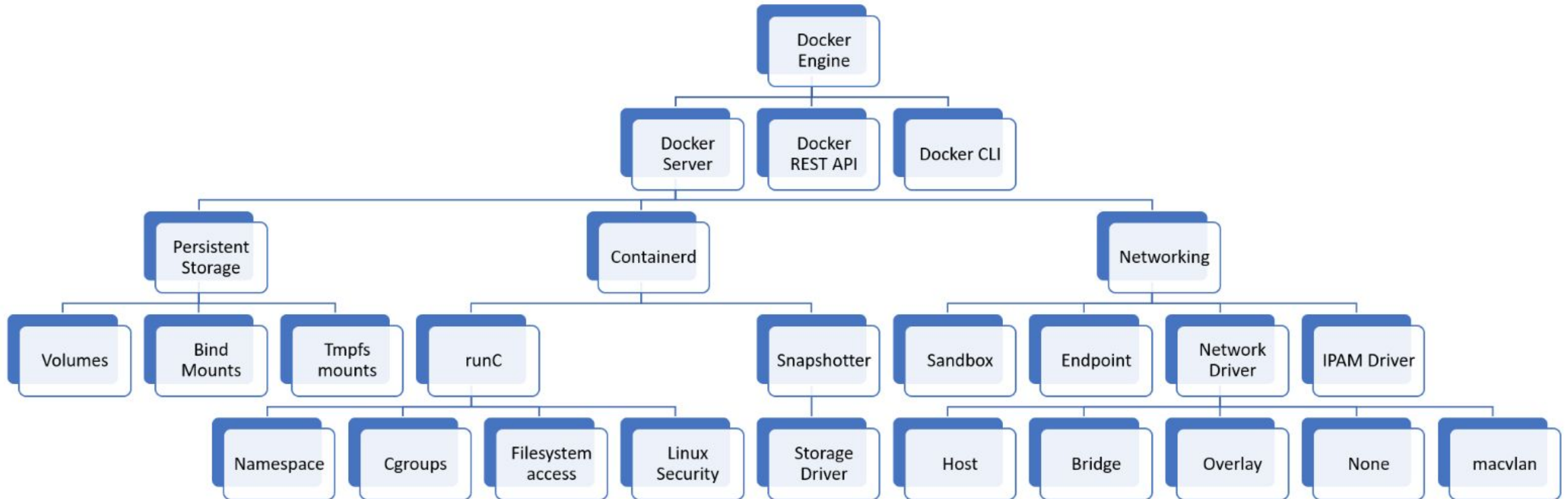
**OCI (Open Container Initiative)**- Linux Foundation project to design open standards for operating-system-level virtualization, most importantly Linux containers.

**CRI-O** - a container runtime

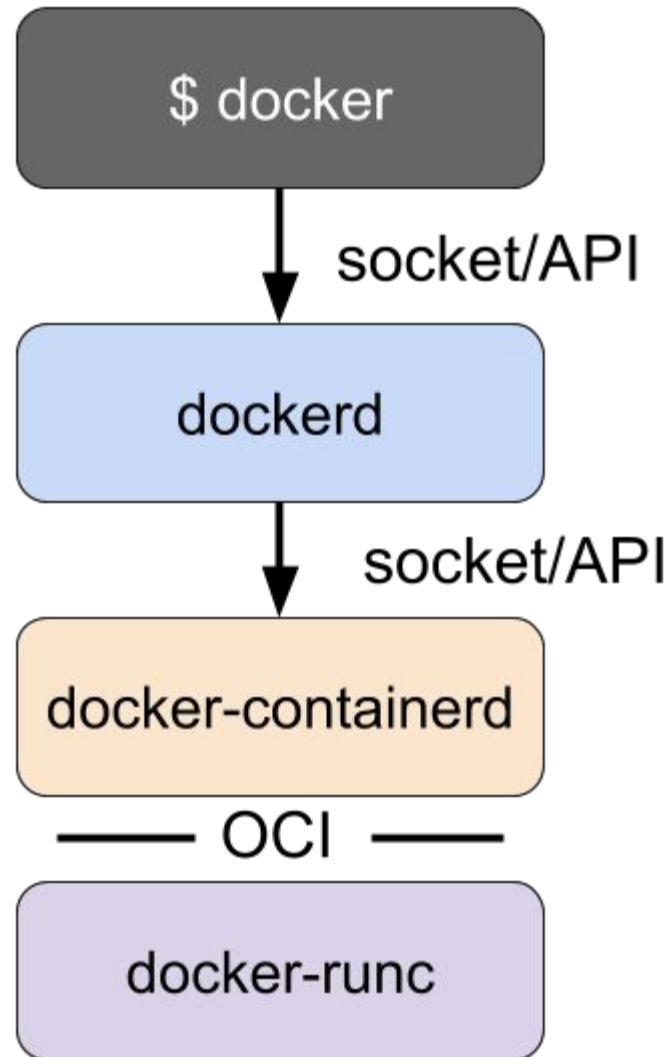
# **Docker, containerd and CRI-O comparison**



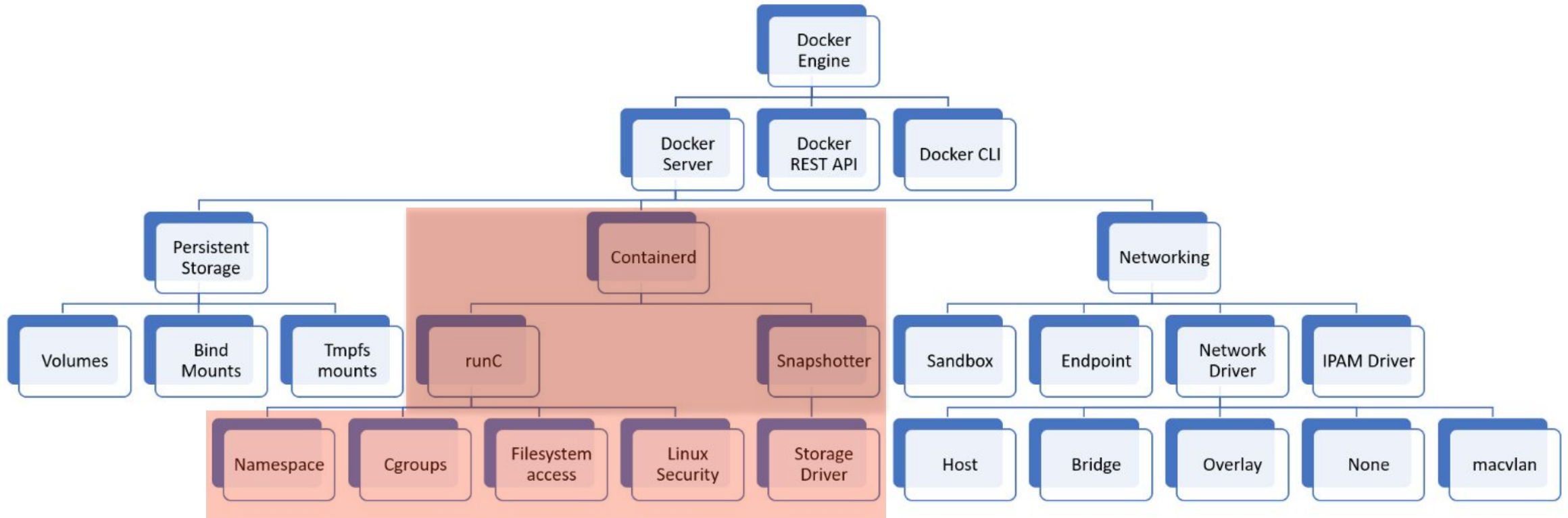
# Docker Architecture



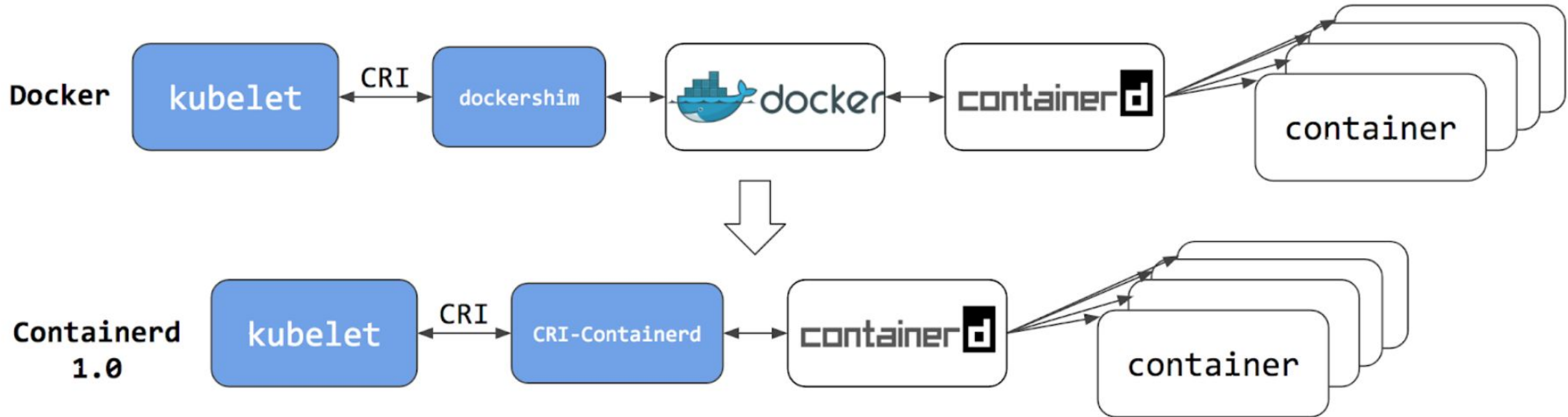
# Simplified Docker architecture after the split



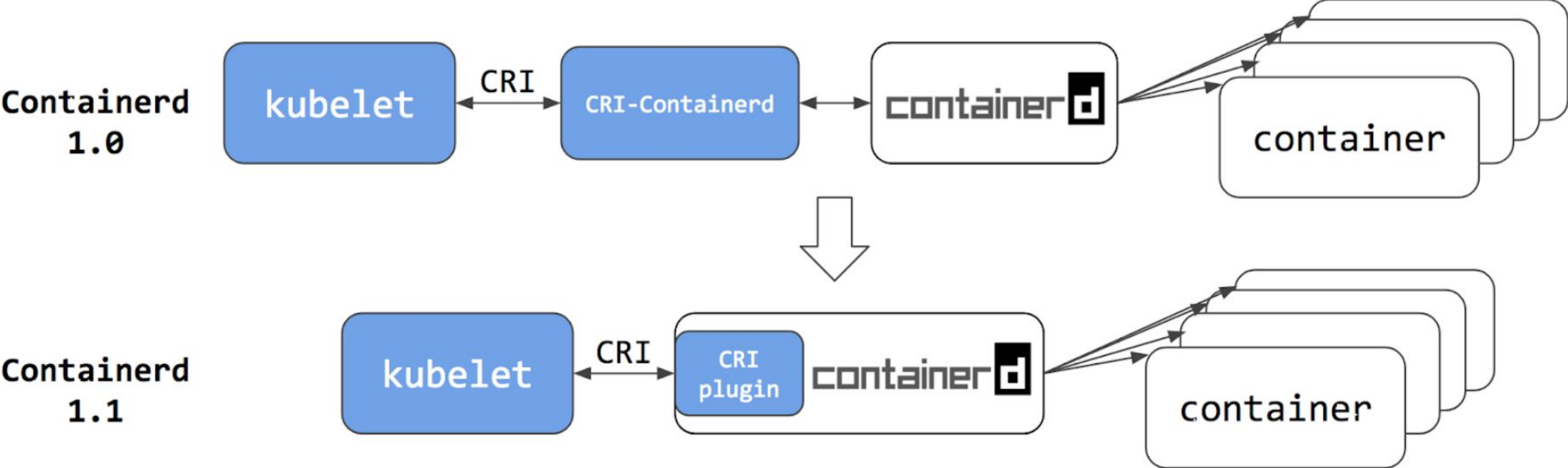
# Docker Architecture



# Docker vs Containerd



# Containerd 1.0 vs Containerd 1.1



# Limitations

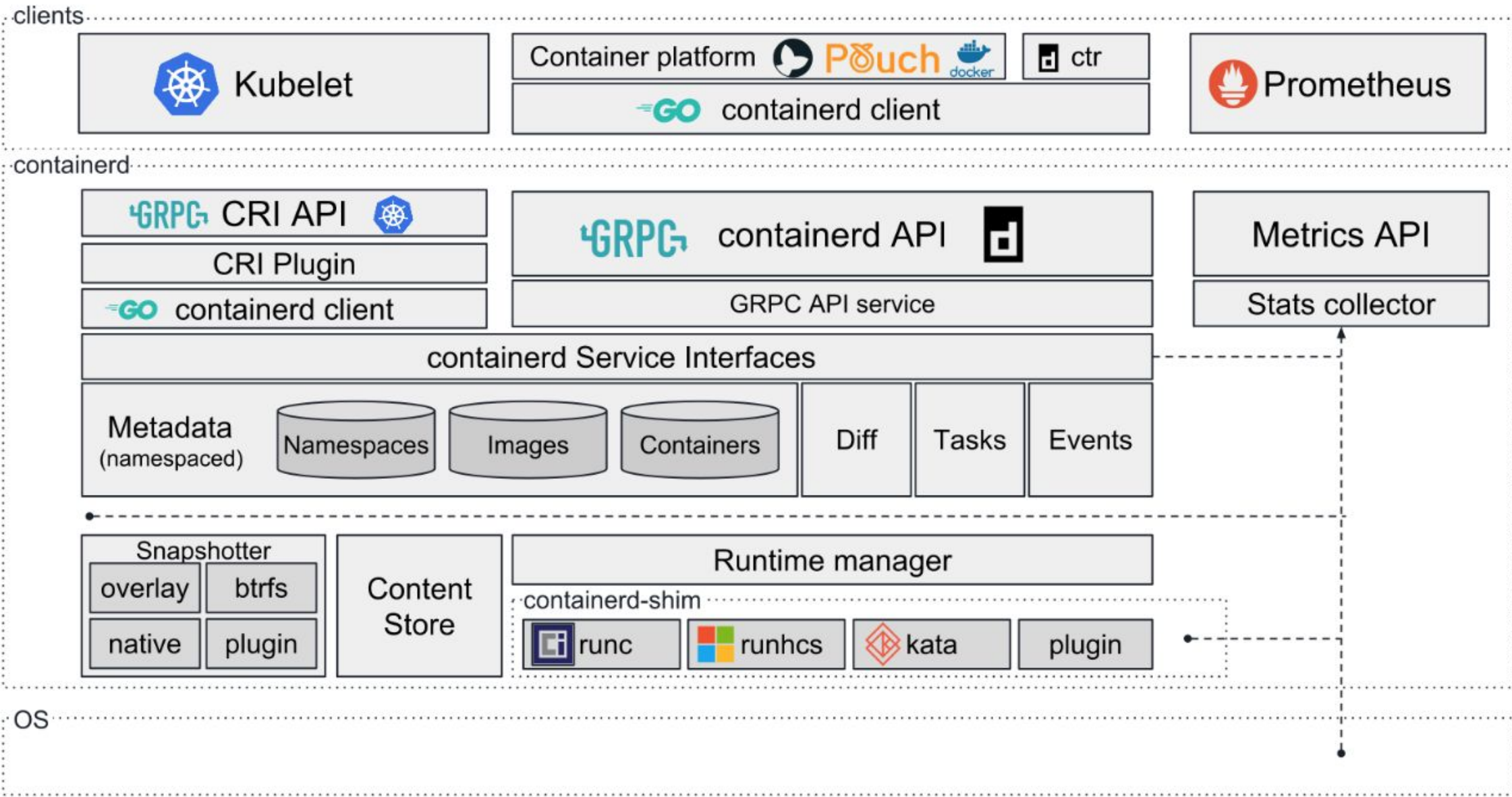
	<i>containerd</i>	<i>Docker Engine</i>
<b>run image in container</b>	<b>x</b>	<b>x</b>
create/use volumes for persistent data		x
build images		x
<b>push/pull images to registry</b>	<b>x</b>	<b>x</b>
interact with/customize container logging		x
commit containers to new image	<i>only via API</i>	x
save/load images	<i>only OCI format</i>	x
create/use virtual bridge networks		x
use network plugins/overlay networks		x
Swarm built-in orchestration		x

# CRI-O

- lightweight CRI runtime made as a Kubernetes specific high-level runtime.
- supports the management of OCI compatible images
- it supports runc and Clear Containers as low-level runtimes
- it supports other OCI compatible low-level runtimes in theory

# Architecture Overview





**Let's talk migration**

# Migration options

- **AWS EKS** doesn't currently support containerd as a runtime
- **Azure AKS** uses Moby as a container runtime
  - customers can deploy their own K8s clusters with a different runtime using aks-engine
- **GKE** - you can add a new node-pool running ``cos_containerd`` and migrate your workloads

# Steps to migrate to `containerd` on GKE

1. Create new node-pools into your cluster using `cos\_containerd` as node image
2. Cordon old node-pools:

```
for node in $(kubectl get nodes -l cloud.google.com/gke-nodepool=default-pool -o=name); do
  kubectl cordon "$node";
done
```

3. Drain pods on the cordoned nodes by running

```
for node in $(kubectl get nodes -l cloud.google.com/gke-nodepool=default-pool -o=name); do
  kubectl drain --force --ignore-daemonsets --delete-local-data --grace-period=10 "$node";
done
```

# User observations



## Troubleshooting

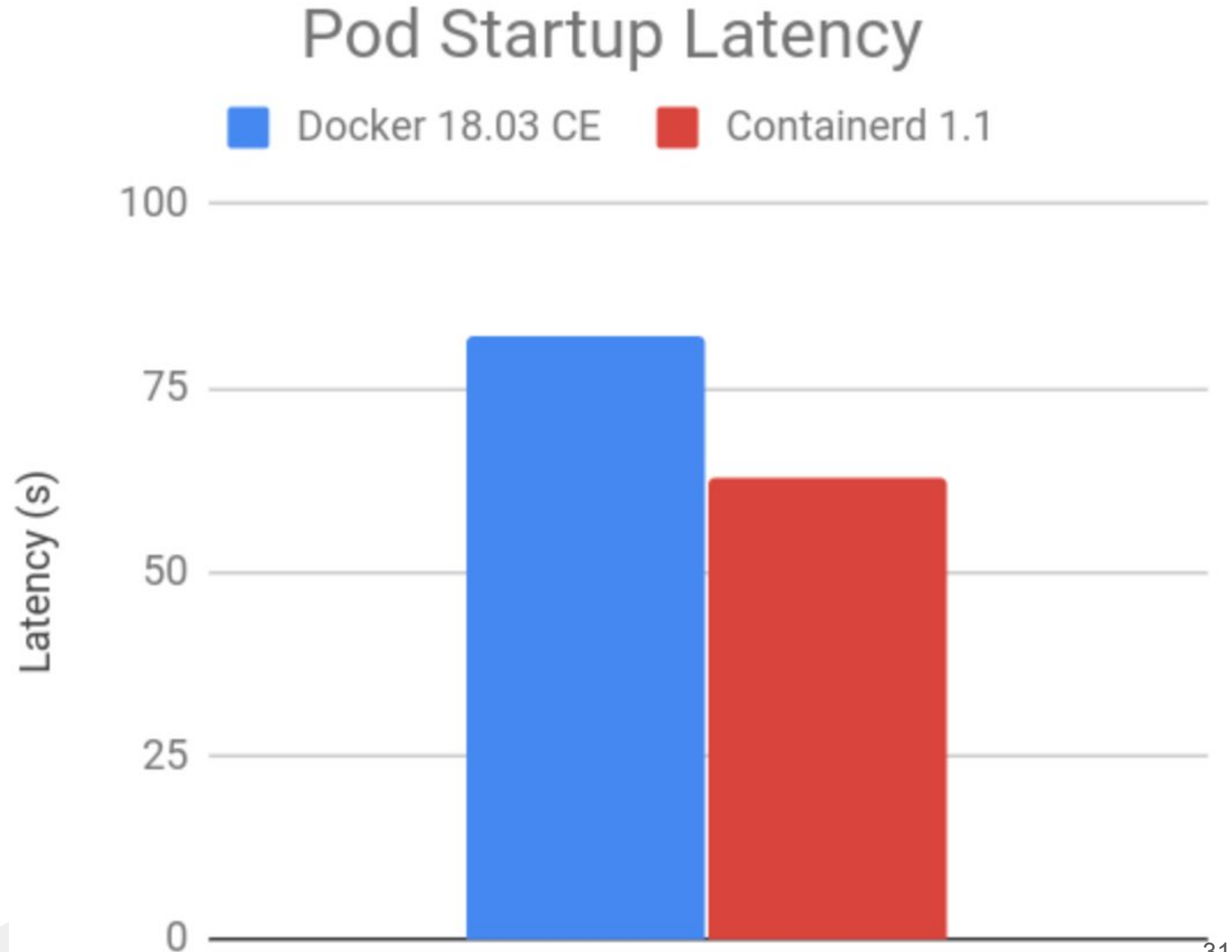
As an SRE with SLAs on my mind  
I care about being able to  
troubleshoot fast:

- `crictl` - troubleshooting containers and pods on nodes in your cluster in an emergency situation
- `ctr` - create and manage containers with containerd, install plugins



# Performance

As a **Systems Engineer** that has to constantly **deploy new features out yesterday** I care about **performance**



# Docker v17.03.2-ce on v1.11.8-gke.6 nodes

```
{
  "containerStatuses": [
    {
      "containerID": "docker://170b55a4d3a8d95102ab79afcdbadb43d0e225c2f82f4341414c2f13f13f786c",
      "image": "k8s.gcr.io/defaultbackend:1.4",
      "imageID": "docker-pullable://k8s.gcr.io/defaultbackend@sha256:865b0c35e6da393b8e80b7e3799f777572399a4cff047eb02a81fa6e7a48ed4b",
      "lastState": {},
      "name": "nginx-ingress-default-backend",
      "ready": true,
      "restartCount": 0,
      "state": {
        "running": {
          "startedAt": "2019-05-15T11:36:09Z"
        }
      }
    }
  ],
  "startTime": "2019-05-15T11:35:49Z"
}
```

shows the container runtime

20 seconds between the time the pod was initialised and the time it was marked ready



# Containerd v1.1.6 on v1.11.8-gke.6 nodes

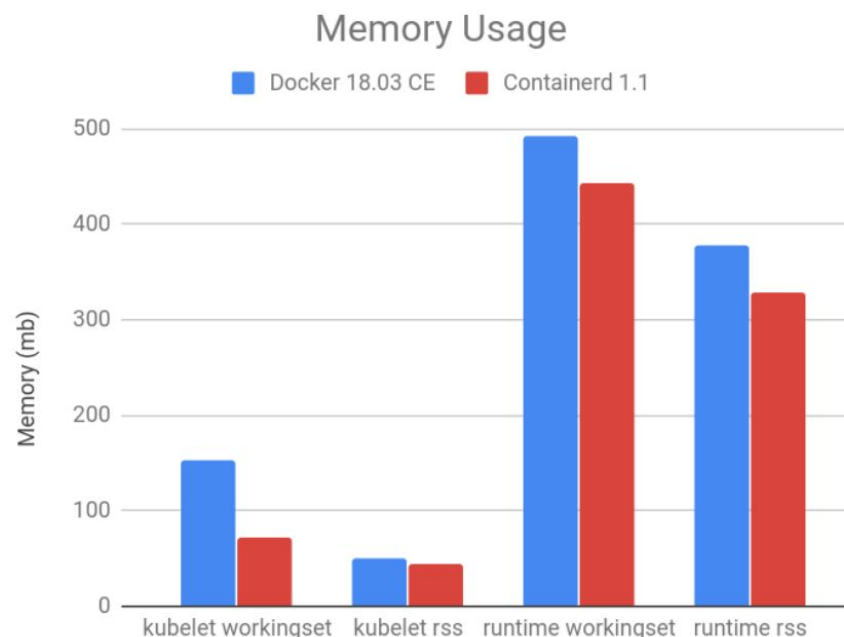
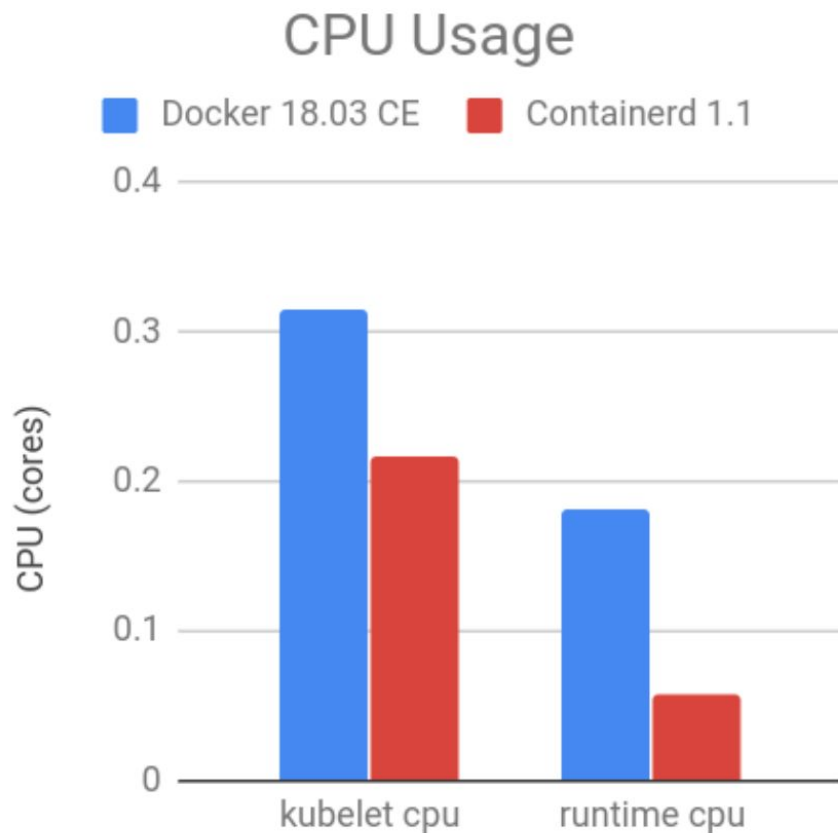
```
{
  "containerStatuses": [
    {
      "containerID": "containerd://4f82b31a116784b75c374db1b40181505872fa649c0aa098984434bf160c30de",
      "image": "k8s.gcr.io/defaultbackend:1.4",
      "imageID": "k8s.gcr.io/defaultbackend@sha256:865b0c35e6da393b8e80b7e3799f777572399a4cff047eb02a81fa6e7a48ed4b",
      "lastState": {},
      "name": "nginx-ingress-default-backend",
      "ready": true,
      "restartCount": 0,
      "state": {
        "running": {
          "startedAt": "2019-05-15T11:35:07Z"
        }
      }
    }
  ],
  "startTime": "2019-05-15T11:35:05Z"
}
```

shows the container runtime

2 seconds between the time the pod was initialised and the time it was marked ready

# Performance

As an Infrastructure Engineer with resource optimization on my mind I care about resource utilisation



# Security

As an engineer with PCI DSS compliance on my mind I care about security

- Containerd has a smaller attack surface than Docker
- they published their full security audit
- with Docker an attacker with access to my nodes can build images locally

# Summary

- Docker's scope is too large for a K8s cluster
- containerd supports multiple low-level runtimes via the "runtime handler" (v1.2)
- containerd runs on top of low level runtimes such as runC, kata-containers
- containerd performs better and is more secure than Docker

# Resources

- ✓ [https://github.com/containerd/containerd/blob/master/docs/SECURITY\\_AUDIT.pdf](https://github.com/containerd/containerd/blob/master/docs/SECURITY_AUDIT.pdf)
- ✓ <http://www.studytrails.com/devops/docker-architecture-engine-containerd-runc/>
- ✓ <https://kubernetes.io/blog/2018/05/24/kubernetes-containerd-integration-goes-ga/>
- ✓ <https://www.ianlewis.org/en/container-runtimes-part-1-introduction-container-r>
- ✓ <https://www.ianlewis.org/en/container-runtimes-part-4-kubernetes-container-r-run>

Current Problems we're  
looking to solve

- Intrusion Detection System
- Service Mesh Setup



PAYBASE\_

We're hiring! Come talk to me or email [engineers@paybase.io](mailto:engineers@paybase.io)

Thank you

We're hiring! Come talk to me or email [engineers@paybase.io](mailto:engineers@paybase.io)

PAYBASE\_