KubeCon | CloudNativeCon

Europe 2018

HYPER.SH

CRI: The Second Boom of Container Runtimes

Xu Wang (@gnawux)

Lei Zhang (@resouer)

Agenda

- CRI Deep Dive
- The Current CRI Implementations
- CRI and Kata Containers

CLOUD NATIVE CON Europe 2017

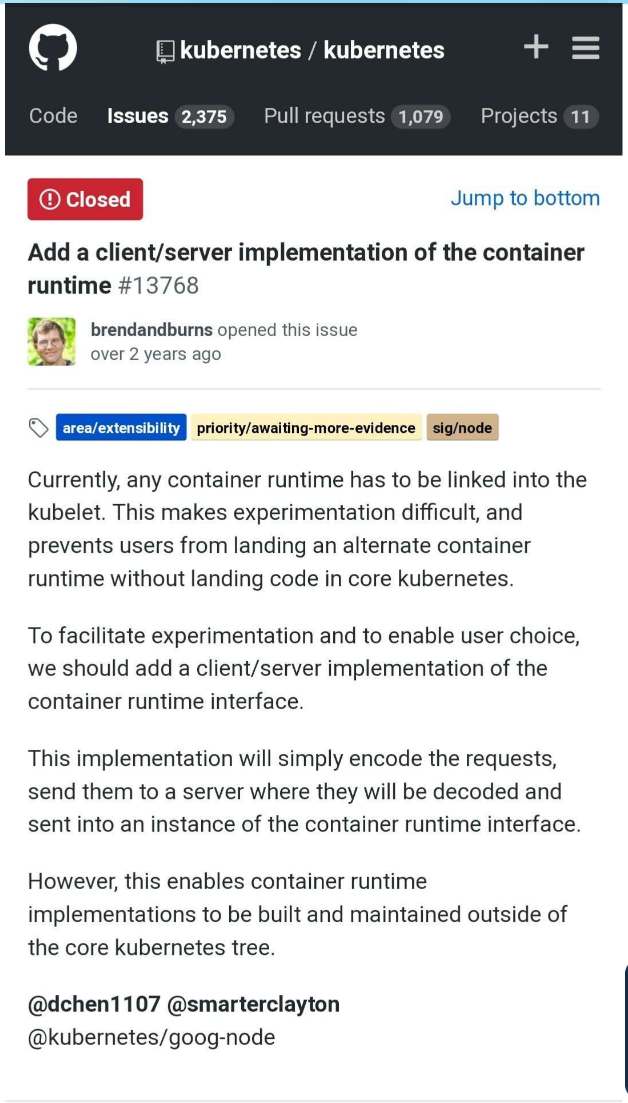
KubeCon A CNCF EVENT

Success of CRI: Bringing Hypervisor based Container to Kubernetes

Harry Zhang, @resouer

A Brief History of CRI

- Once upon a time…
  - rkt was added into kubelet as the 2nd runtime.
    - Increased the complexity on maintenance
  - Docker (the 1st runtime) introduced more and more feature.
    - Don’t like a simple runtime any more
  - Hyper.sh joined the community and tried to become a third runtime.



Add a client/server implementation of the container runtime #13768

brendandburns opened this issue over 2 years ago

area/extensibility priority/awaiting-more-evidence sig/node

Currently, any container runtime has to be linked into the kubelet. This makes experimentation difficult, and prevents users from landing an alternate container runtime without landing code in core kubernetes.

To facilitate experimentation and to enable user choice, we should add a client/server implementation of the container runtime interface.

This implementation will simply encode the requests, send them to a server where they will be decoded and sent into an instance of the container runtime interface.

However, this enables container runtime implementations to be built and maintained outside of the core kubernetes tree.

@dchen1107 @smarterclayton
@kubernetes/goog-node

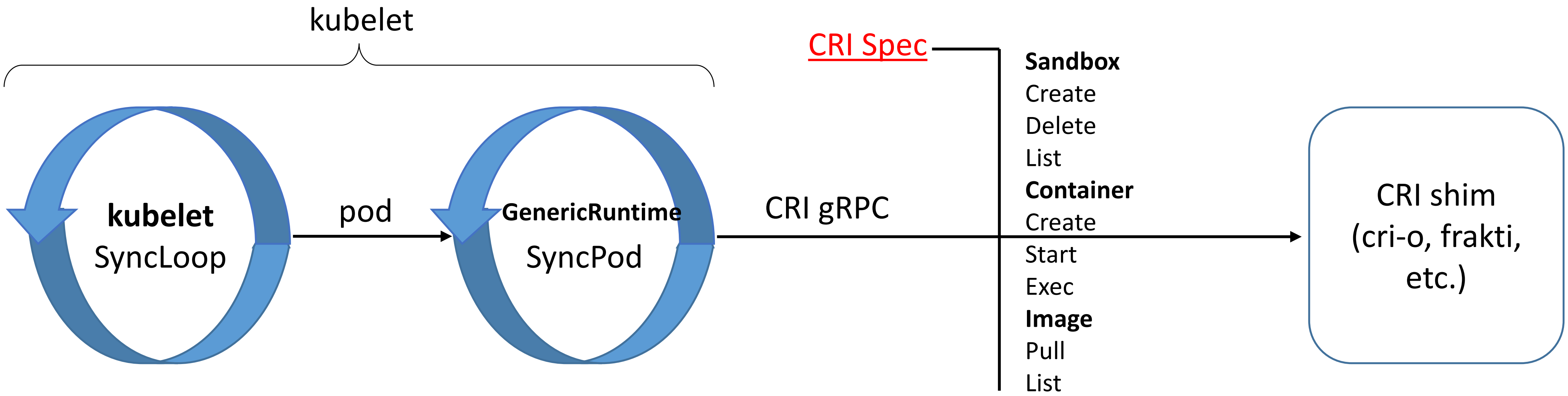
The Birth of CRI

The Kubelet should not vendor a runtime implementation.

(Unfortunately, this is not the original words literally)

- Developers from Google, CoreOS, and Hyper.sh drafted a kubelet-runtime interface together.
- The interface, CRI, was written with gRPC
  - gRPC had already been open sourced at that time.
  - The performance difference between gRPC and HTTP/REST was tested
- First CRI implementation: dockershim
- First Non-Docker CRI implementation: Frakti

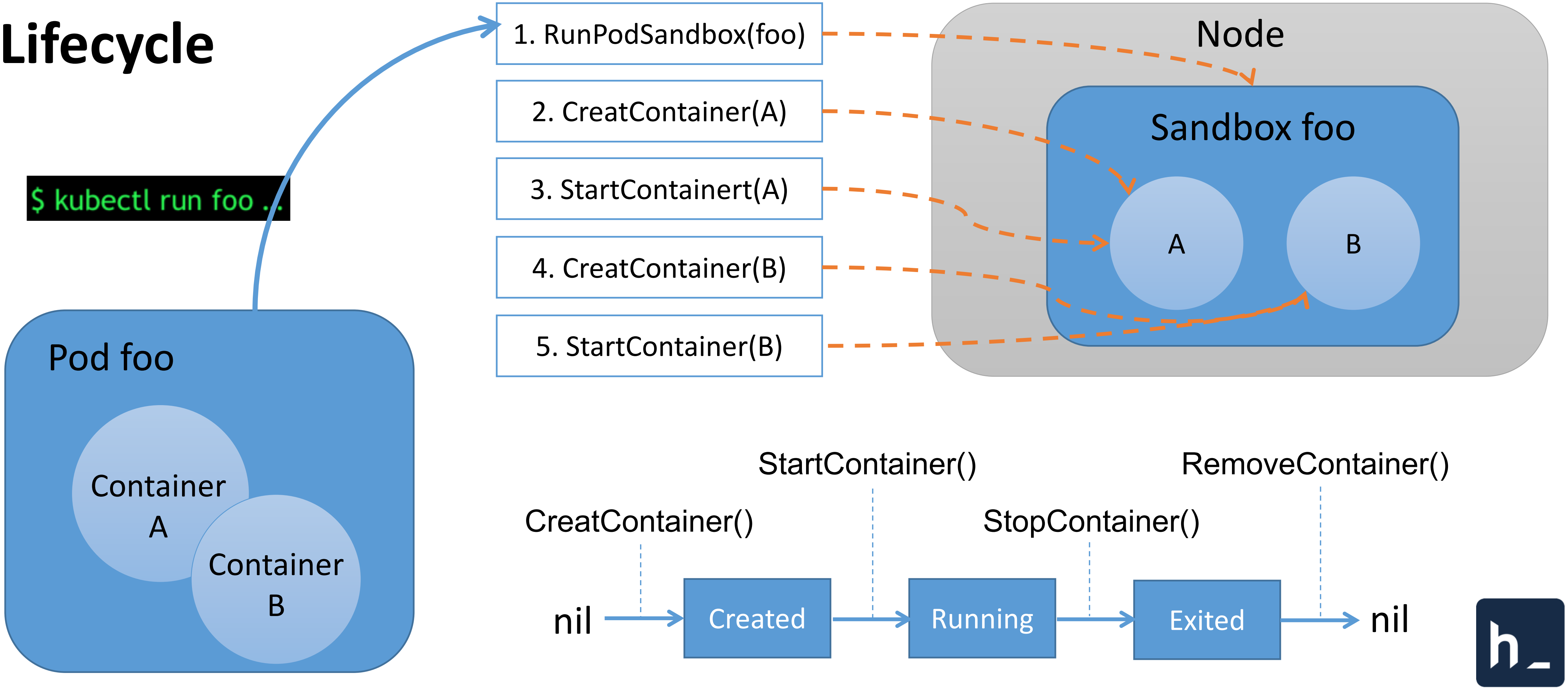
The CRI Interface



- Describe what kubelet expects from container runtimes
- Imperative container-centric interface
- Extensibility

Implement a CRI Runtime (1)

Lifecycle

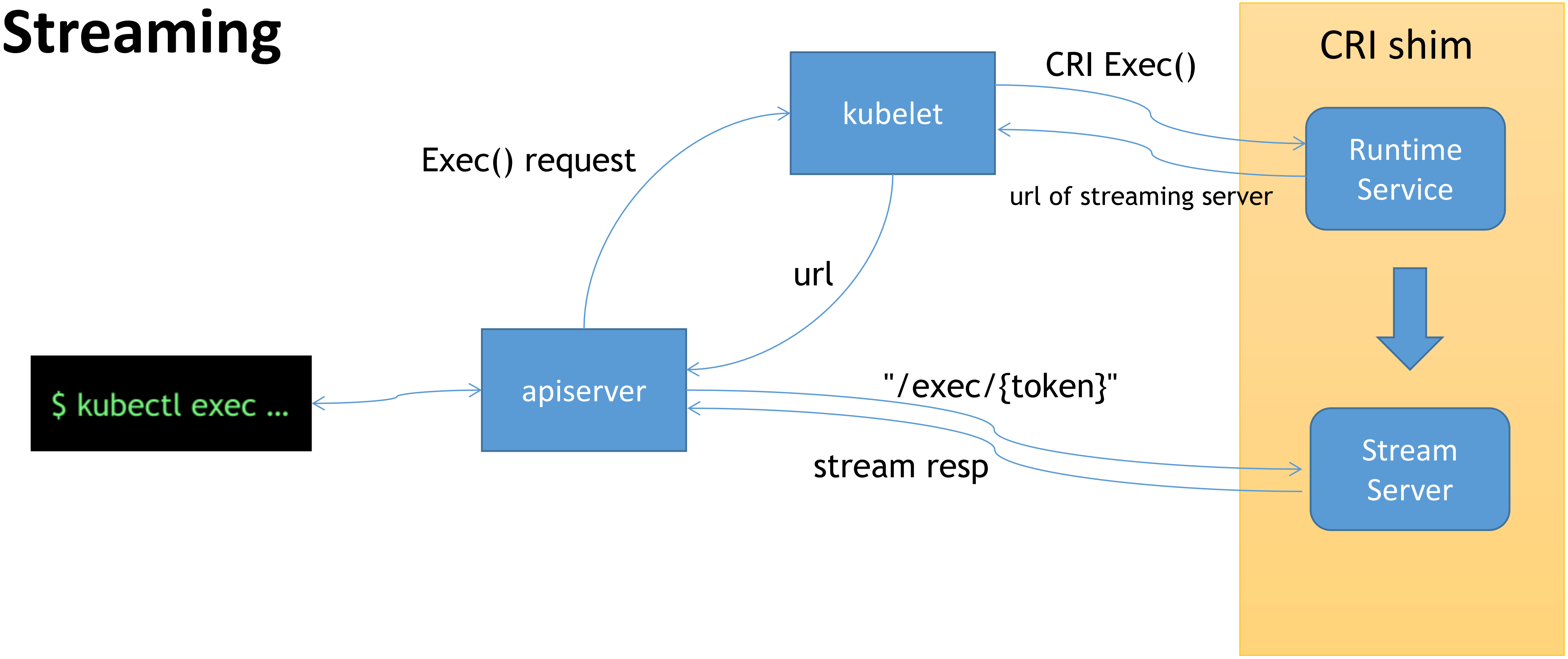


```
$ kubectl run foo ...
```

Implement a CRI Runtime (2)

KubeCon CloudNativeCon Europe 2018

Streaming



KubeCon | CloudNativeCon

Europe 2018

HYPER.SH

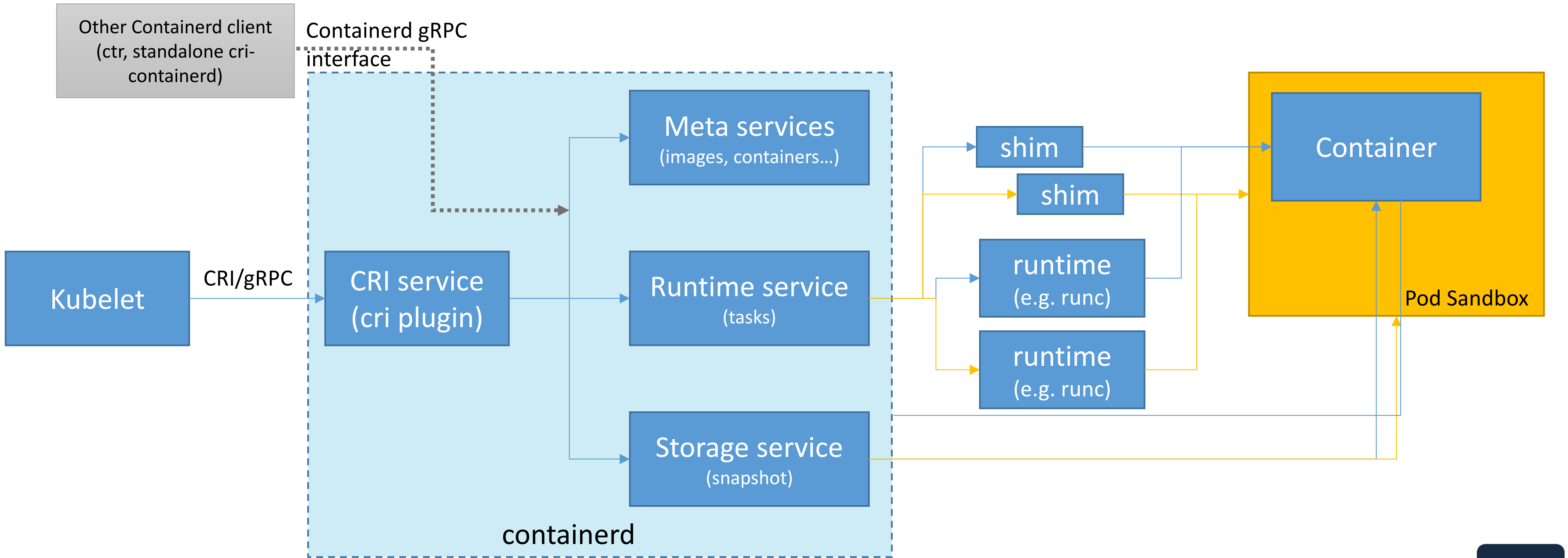
The Existing CRI Runtimes

CRI-O, cri-containerd, frakti, and how they work

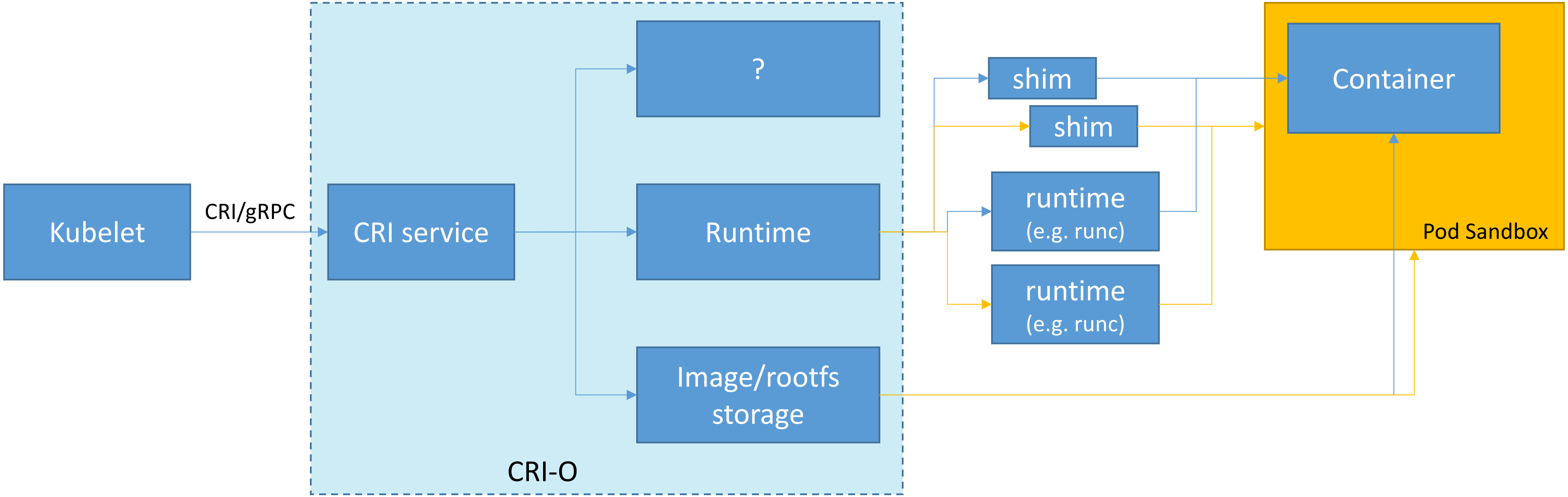
The Existing CRI Runtimes

- containerd/cri (cri-containerd)
  - containerd and k8s node team
- CRI-O
  - Red Hat etc.
- frakti
  - Hyper.sh etc.

Containerd (w/ cri plugin) in brief

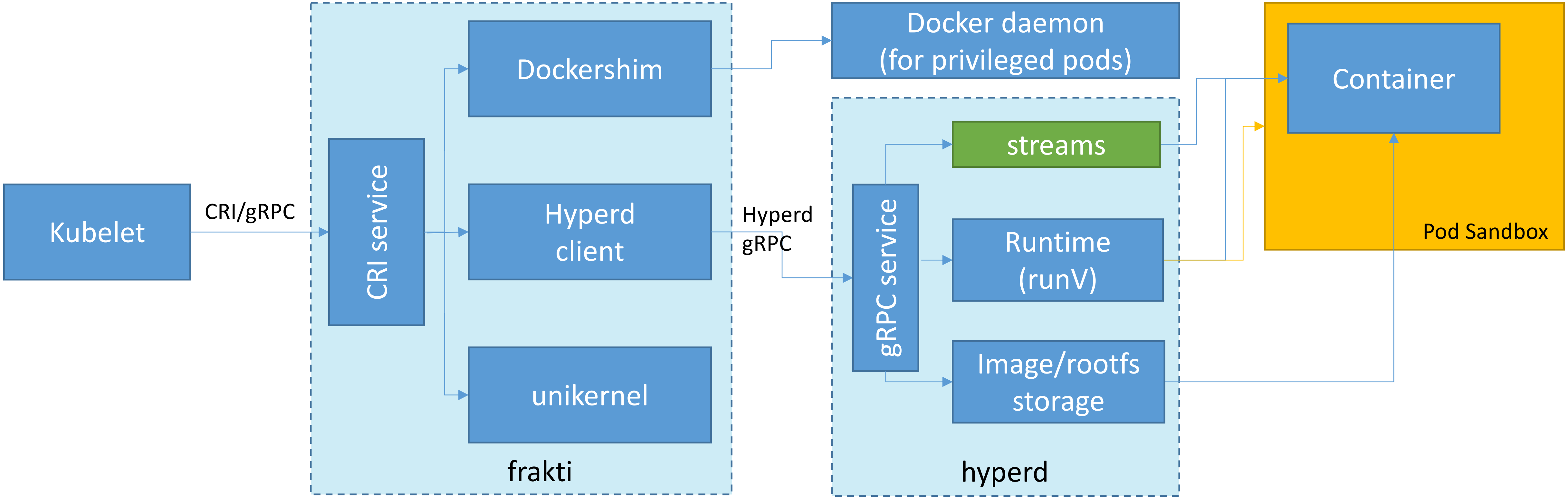

Other Containerd client (ctr, standalone cri-containerd)
Containerd gRPC interface
Kubelet
CRI/gRPC
CRI service (cri plugin)
Meta services (images, containers…)
Runtime service (tasks)
Storage service (snapshot)
containerd
shim
shim
runtime (e.g. runc)
runtime (e.g. runc)
Container
Pod Sandbox

CRI-O in brief



Similar to Containerd, right?

Frakti 1.0 in brief



- Frakti is a CRI shim
- Hyperd implements the CRI semantics for runV

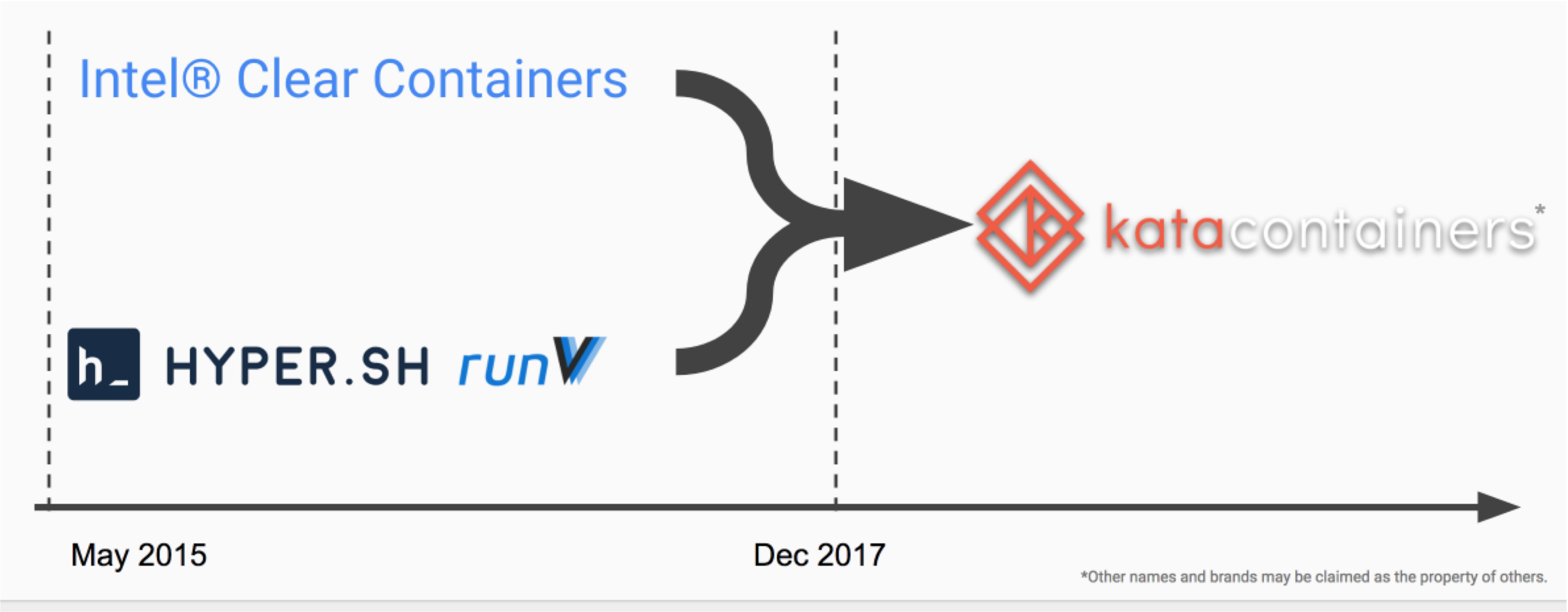
KubeCon | CloudNativeCon

Europe 2018

HYPER.SH

CRI and Kata Containers

History of Kata



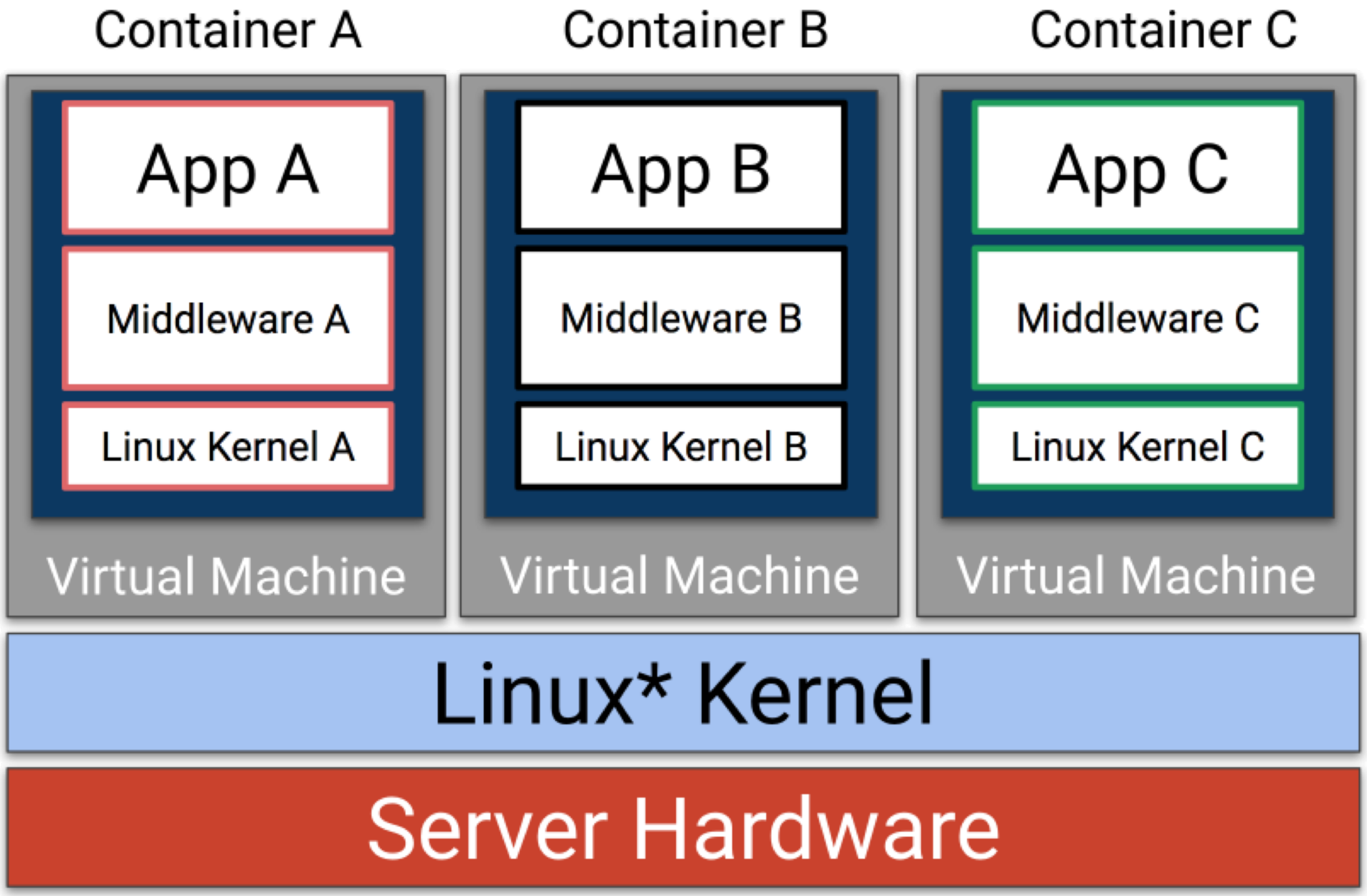
*Other names and brands may be claimed as the property of others.

Technical Vision of Kata

- Light and fast VM-based containers
- Merge Intel® Clear Containers and Hyper runV technologies
- Seamless integration with Kubernetes (CRI), Docker and Openstack
- Support multiple architectures
- Support multiple hypervisors

How Kata Works

- Each container/pod is hypervisor isolated
- As secure as a VM
- As fast as a container
- Seamless integration with the container ecosystem and management layers

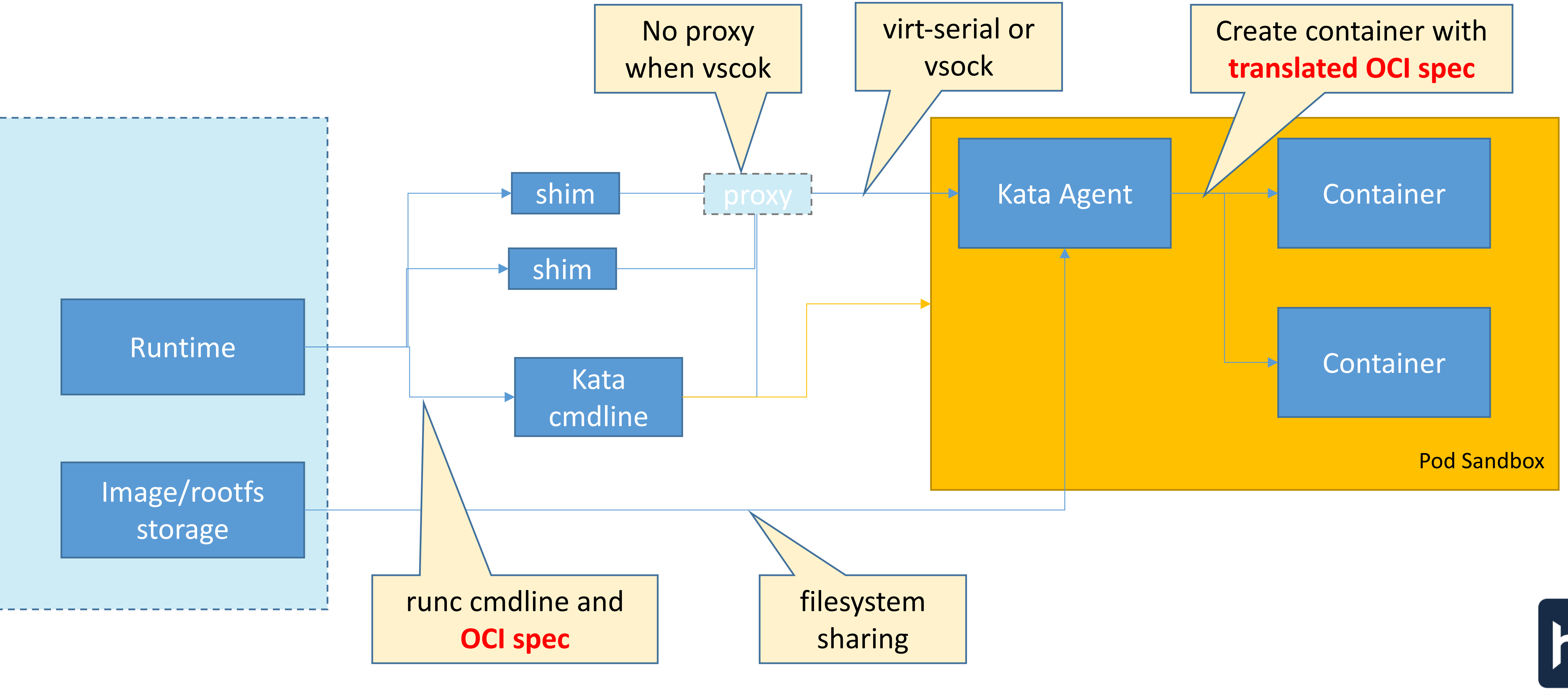


*Other names and brands may be claimed as the property of others.

CRI for Kata Containers

- Kata Containers is designed to be compatible with Kubernetes
- Integration Methods
  - Kata Containers as a standalone tool
    - runC compatible command line
    - OCI specs
    - Work with containerd and CRI-O
  - Kata Containers as a runtime lib
    - runV compatible APIs
    - Work with frakti
    - (containerd plugin?)

containerd/CRI-O + Kata



No proxy when vscok

virt-serial or vsock

Create container with **translated OCI spec**

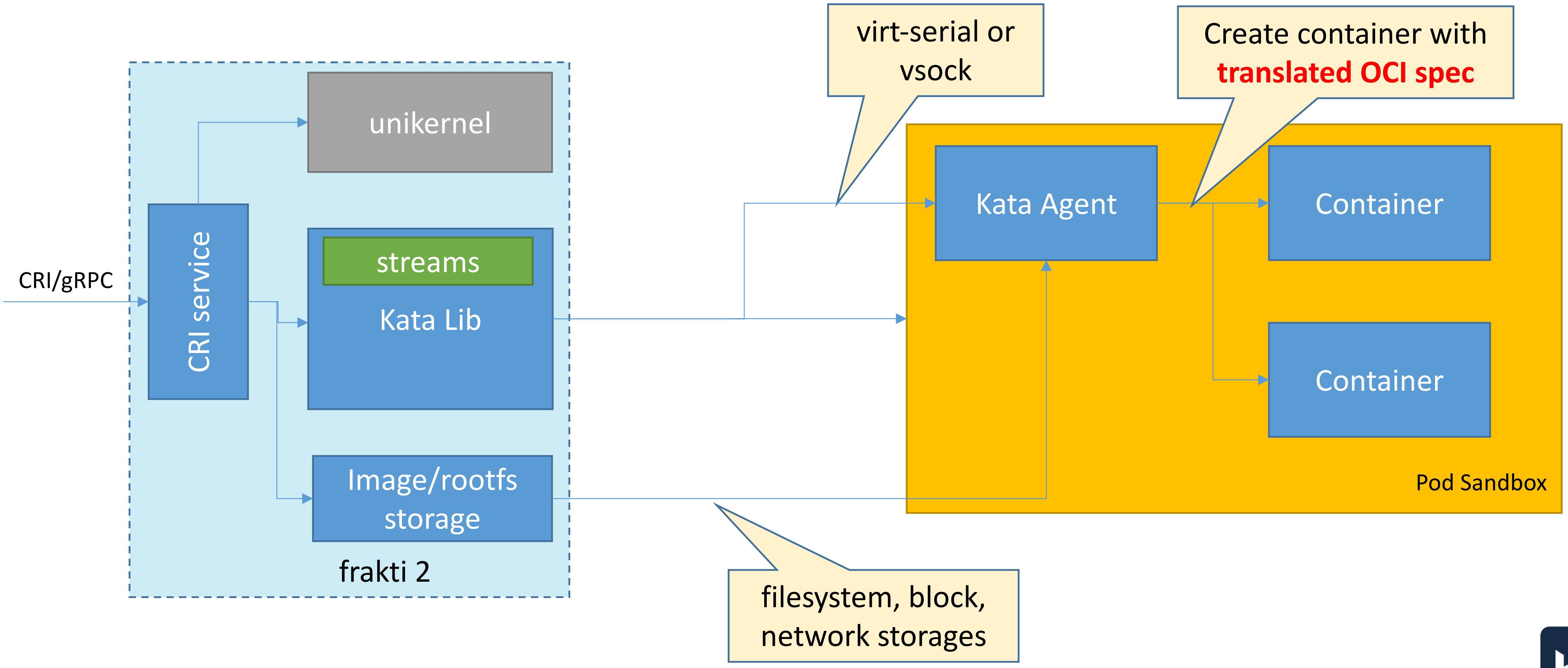
runc cmdline and **OCI spec**

filesystem sharing

Frakti + Kata

KubeCon | CloudNativeCon

Europe 2018



Comparison

- Share more code parts
- Shims and Proxy Process
  - Process
  - Memory
- And more resources
  - Rootfs and Volumes
  - Networking plug-ins

KubeCon | CloudNativeCon

Europe 2018

HYPER.SH

Thank You