KubeCon | CloudNativeCon

North America 2018

KataContainers the Hard Way:

Kubernetes + containerd + KataContainers

The Speakers

- Lei Zhang (@resouer)
  - Co-maintaining Kubernetes
  - Contributor of Kata Containers
  - Alibaba Group
  - Formerly hyper.sh
- Xu Wang (@gnawux)
  - Kata Containers Arch Committee
  - CTO Co-Founder of hyper.sh

Agenda

- Kata Containers: a General Introduction
- Kubernetes CRI: Enabling the Pluggability of Container Runtimes
- Connect Kata Containers & Kubernetes
- Future works

What’s Kata Containers

- A container runtime, like runC
- Built w/ virtualization tech, like VM
- Initiated by hyper.sh and Intel®
- Hosted by OpenStack Foundation
- Contributed by Huawei, Google, MSFT, etc.

kata containers

Kata Containers is Virtualized Container

Combine the Best from Both V* and C*

kata containers

- Industry Compatibility
- Enhanced Networking
- Direct Device Support
- Run Custom Kernel
- More Secure

From runV to Kata Containers

Intel®
Clear Containers

HYPER.SH
runV

katacontainers
Announced project

Release 1.0

May 2015

Dec 2017

May 2018

Recent Developments

- v1.4.0 was just released late November (https://github.com/kata-containers/runtime/releases/tag/1.4.0)
  - 6-week release schedule
  - Hotplug improve
  - Template/Factory from runV
  - VSock support
  - Multi-Arch support
  - Improved devices support

$\Big\}$ Highlight features from 1.1.0-1.4.0

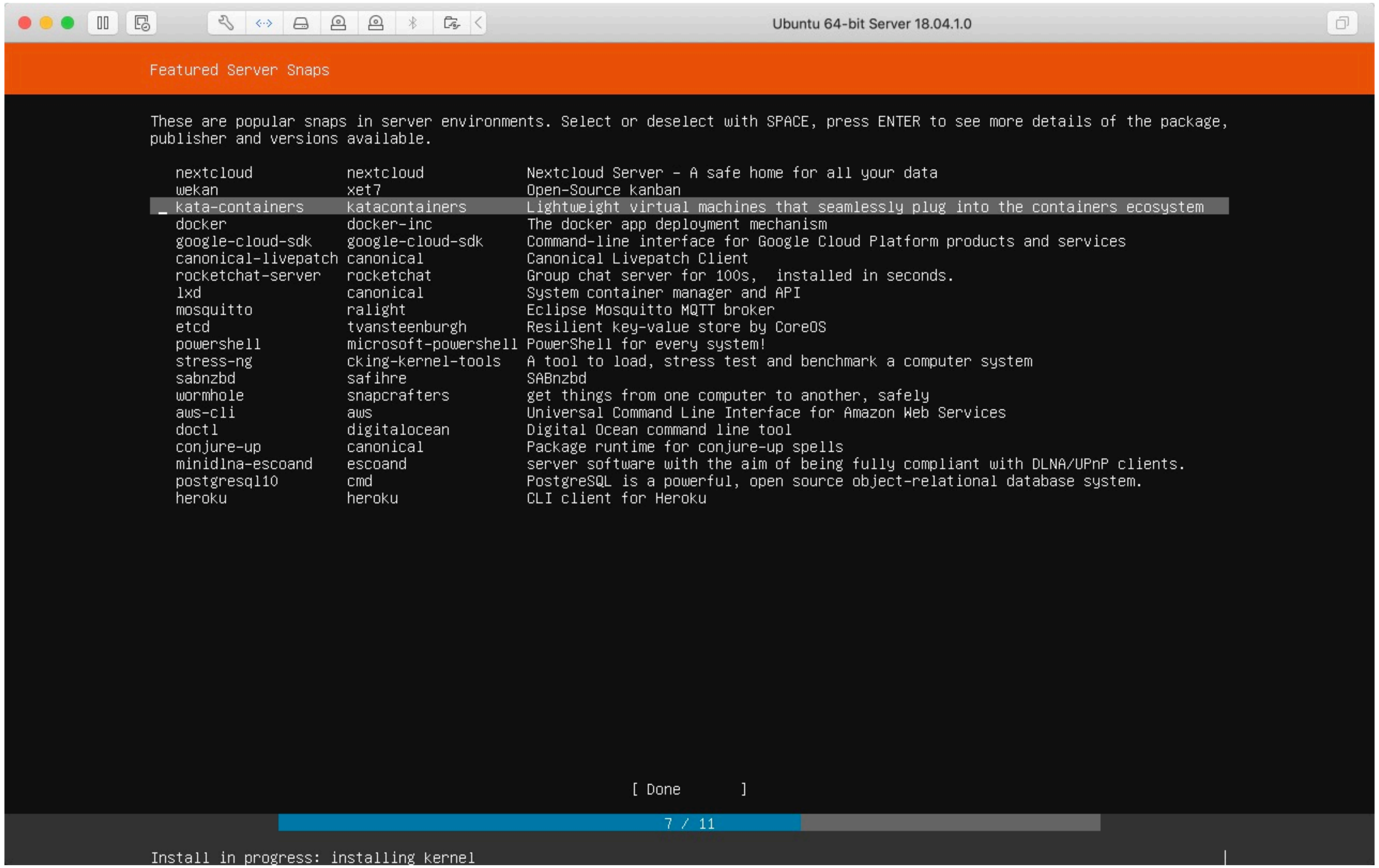
- Ongoing development
  - Containerd Shim V2 Support ← Merged and will be in 1.5.x
  - Nemu as a VMM
  - More on hotplug
  - Live upgrading
  - Etc.

Now Let’s Start

```
$ ARCH=$(arch)
$ sudo sh -c "echo 'deb
http://download.opensuse.org/repositories/home:/katacontainers:/releases:/${A
RCH}:/master/xUbuntu_$(lsb_release -rs)/ /' > /etc/apt/sources.list.d/kata-
containers.list"
$ curl -sL
http://download.opensuse.org/repositories/home:/katacontainers:/releases:/${A
RCH}:/master/xUbuntu_$(lsb_release -rs)/Release.key | sudo apt-key add -
$ sudo -E apt-get update $ sudo -E apt-get -y install kata-runtime kata-proxy
kata-shim
```

https://github.com/kata-containers/documentation/blob/master/install/ubuntu-installation-guide.md

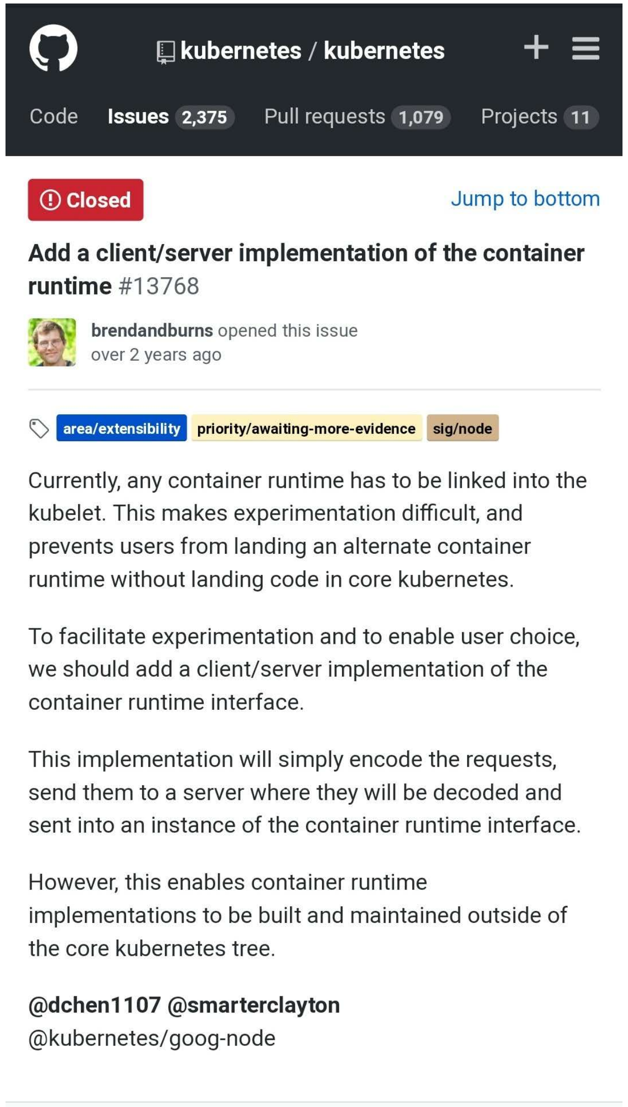
Or may be even simpler…



```
Ubuntu 64-bit Server 18.04.1.0

Featured Server Snaps

These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see more details of the package,
publisher and versions available.

   nextcloud            nextcloud            Nextcloud Server - A safe home for all your data
   wekan                xet7                 Open-Source kanban
 _ kata-containers      katacontainers       Lightweight virtual machines that seamlessly plug into the containers ecosystem
   docker               docker-inc           The docker app deployment mechanism
   google-cloud-sdk     google-cloud-sdk     Command-line interface for Google Cloud Platform products and services
   canonical-livepatch  canonical            Canonical Livepatch Client
   rocketchat-server    rocketchat           Group chat server for 100s,  installed in seconds.
   lxd                  canonical            System container manager and API
   mosquitto            ralight              Eclipse Mosquitto MQTT broker
   etcd                 tvansteenburgh       Resilient key-value store by CoreOS
   powershell           microsoft-powershell PowerShell for every system!
   stress-ng            cking-kernel-tools   A tool to load, stress test and benchmark a computer system
   sabnzbd              safihre              SABnzbd
   wormhole             snapcrafters         get things from one computer to another, safely
   aws-cli              aws                  Universal Command Line Interface for Amazon Web Services
   doctl                digitalocean         Digital Ocean command line tool
   conjure-up           canonical            Package runtime for conjure-up spells
   minidlna-escoand     escoand              server software with the aim of being fully compliant with DLNA/UPnP clients.
   postgresql10         cmd                  PostgreSQL is a powerful, open source object-relational database system.
   heroku               heroku               CLI client for Heroku

                                   [ Done       ]

                                       7 / 11

Install in progress: installing kernel
```

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

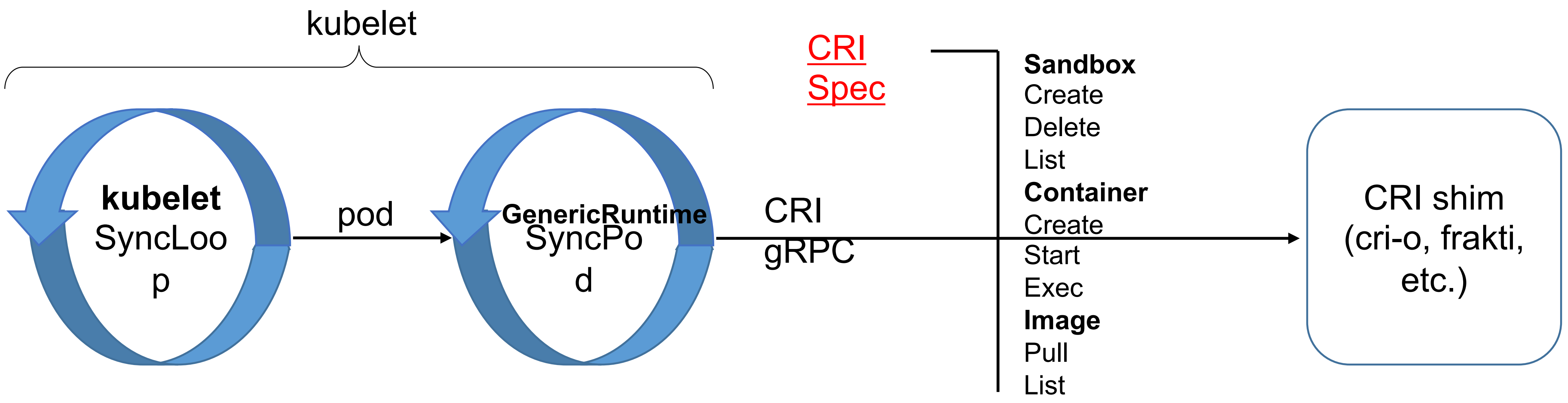
KubeCon | CloudNativeCon
North America 2018

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

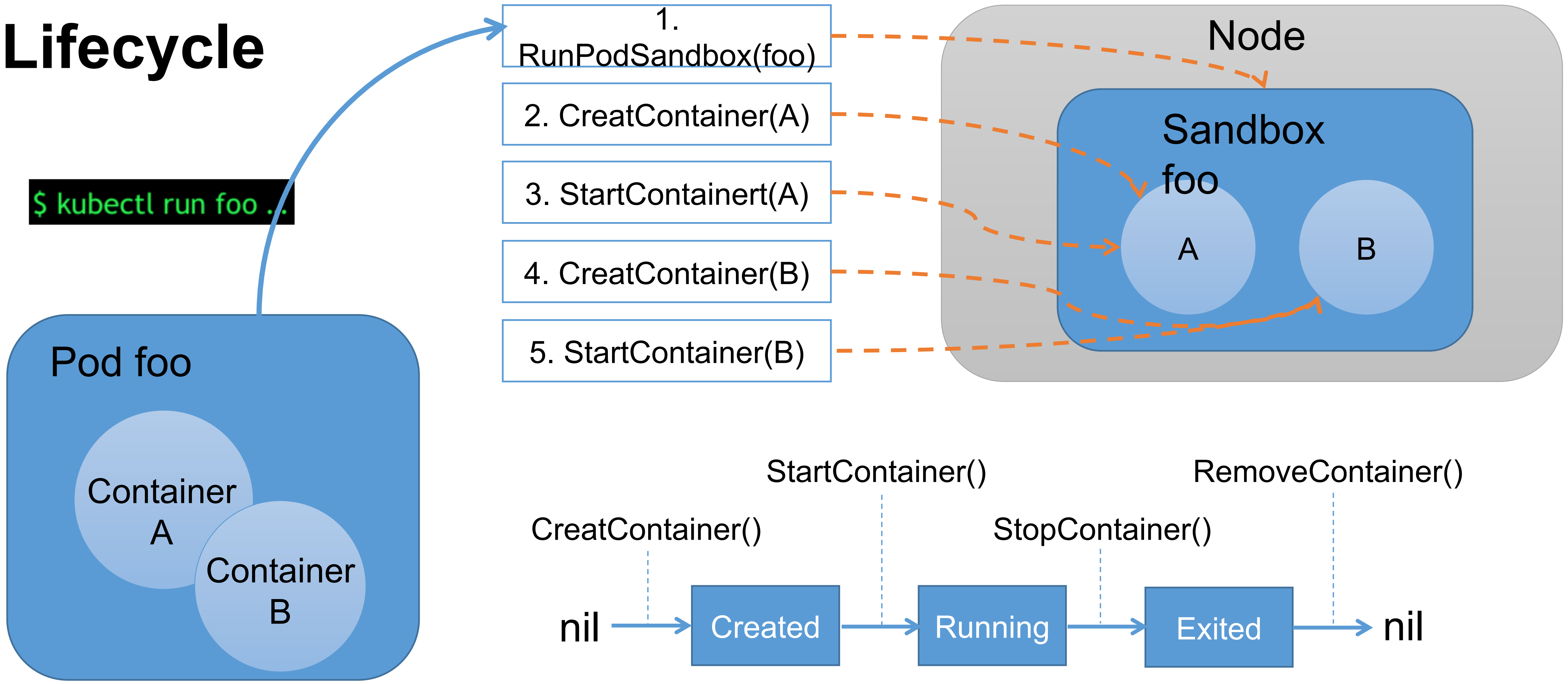
The CRI Interface



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

Implement a CRI Runtime (1)

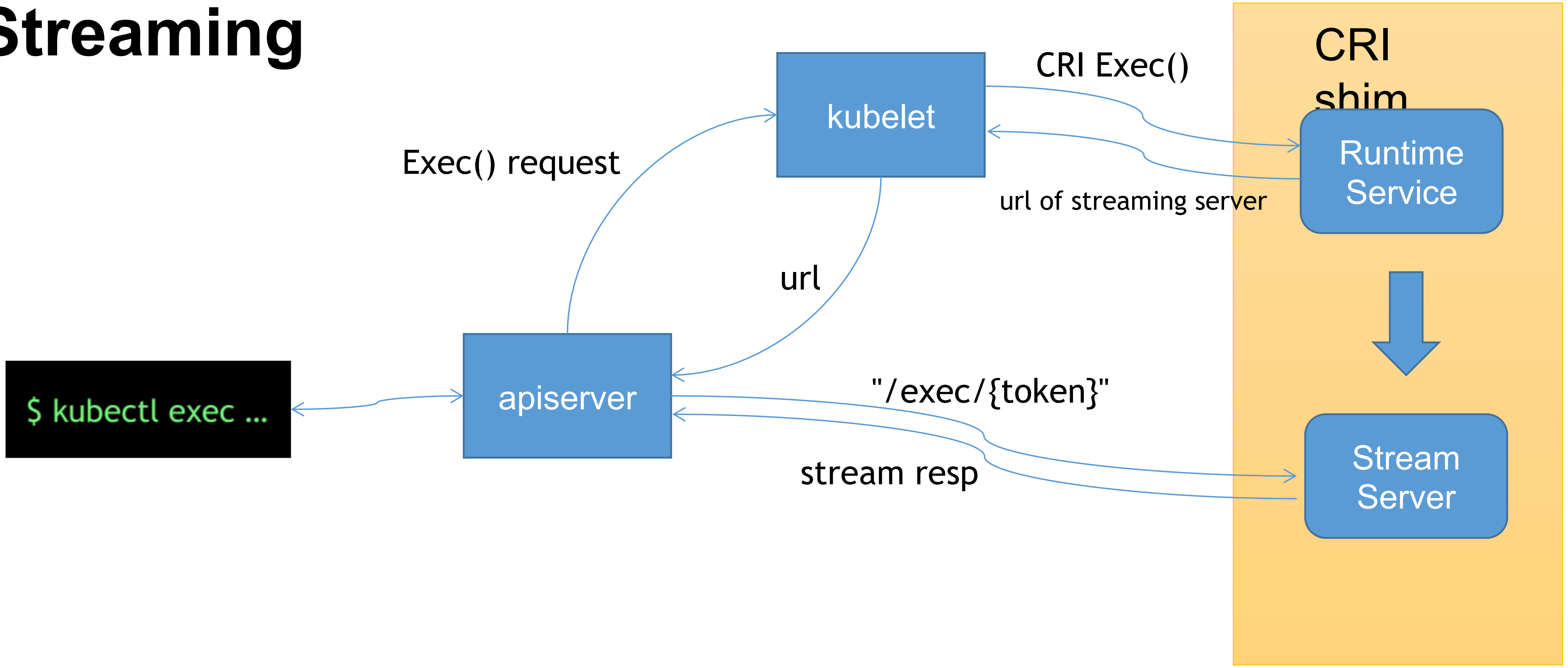
Lifecycle



Implement a CRI Runtime (2)

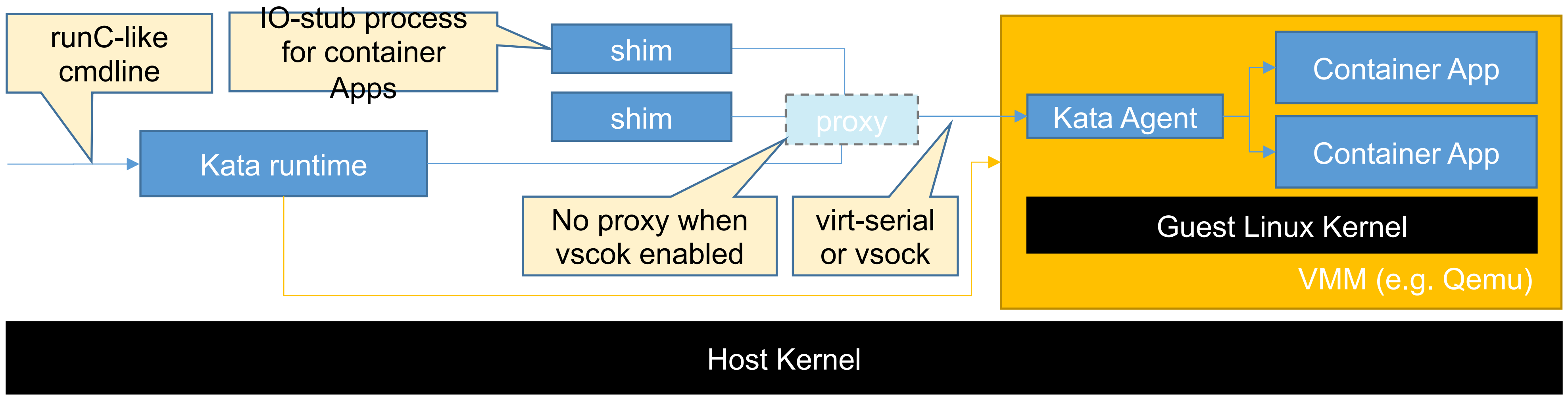
KubeCon | CloudNativeCon North America 2018

Streaming



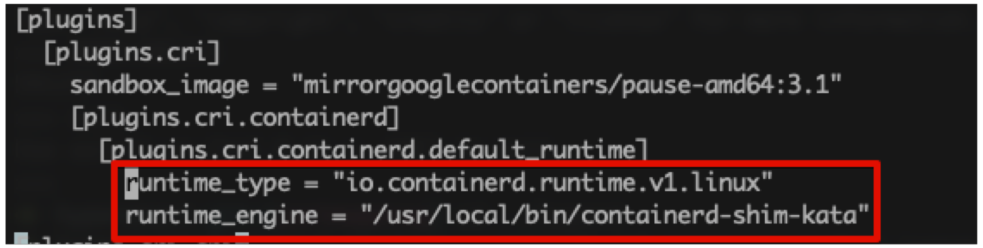
Run Kata Containers w/ Docker

- Just put it on the position of runC

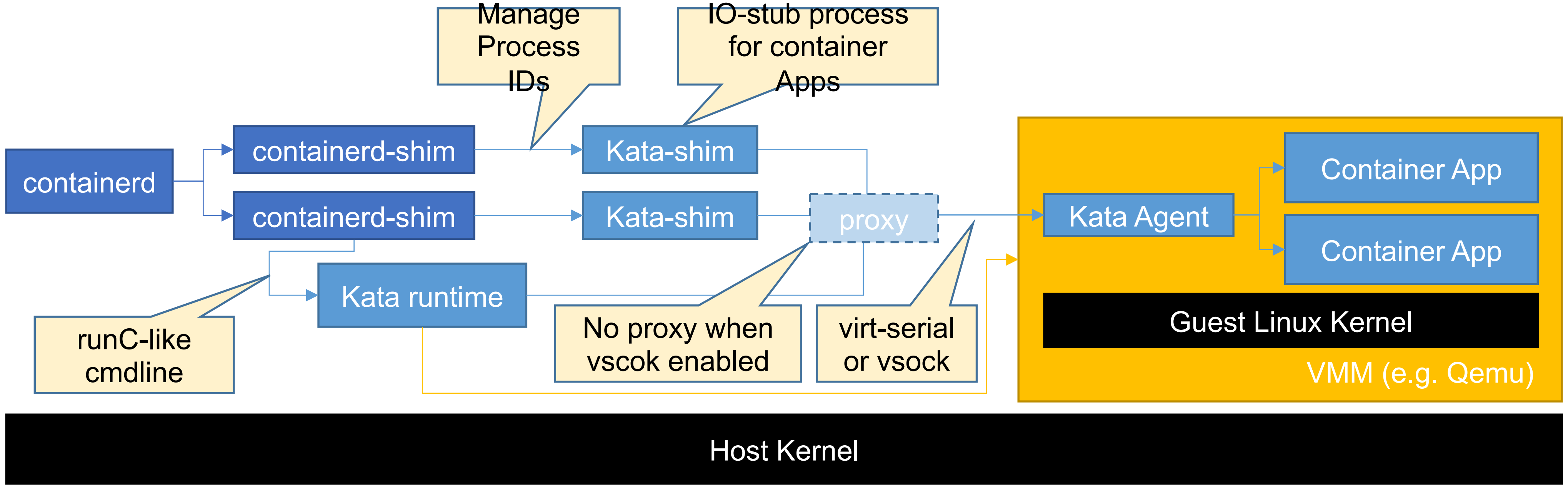


Run Kata Containers (pre-1.5) w/ k8s

- It works
- But looks not so elegent…

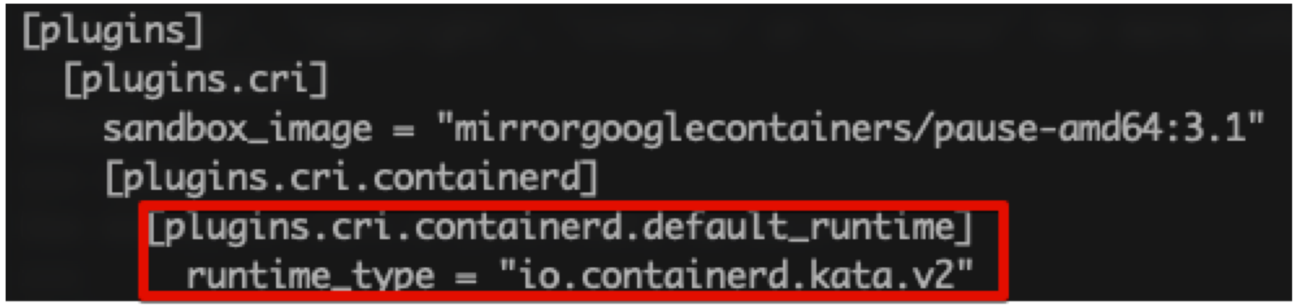


```
[plugins]
  [plugins.cri]
    sandbox_image = "mirrorgooglecontainers/pause-amd64:3.1"
    [plugins.cri.containerd]
      [plugins.cri.containerd.default_runtime]
        runtime_type = "io.containerd.runtime.v1.linux"
        runtime_engine = "/usr/local/bin/containerd-shim-kata"
```

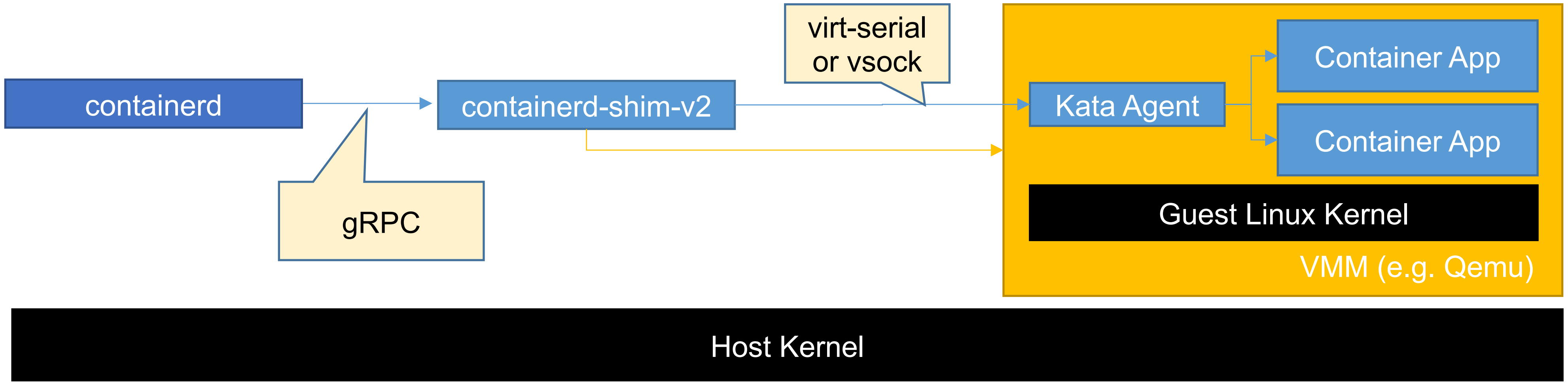


Run Kata Containers (post-1.5) w/ k8s

- Then much simpler



```
[plugins]
  [plugins.cri]
    sandbox_image = "mirrorgooglecontainers/pause-amd64:3.1"
    [plugins.cri.containerd]
      [plugins.cri.containerd.default_runtime]
        runtime_type = "io.containerd.kata.v2"
```



KubeCon | CloudNativeCon

North America 2018

Hands-on Time

Related and Future Works

- About CRI-O Support
  - Similar and different parts
- About Storage
  - Block or filesystem
- About Networking
  - Common CNI plugins support
  - Optimized Networking support

Contribute

- Website: https://katacontainers.io
- Code and documentation hosted on https://github.com/kata-containers/
- Major releases managed through Github* Projects
- Intel (Intel® Clear Containers) & Hyper.sh (runV) contributing initial IP
- Apache 2 license
- Slack: katacontainers.slack.com
- IRC: #kata-dev@freenode
- Mailing-list: kata-dev@lists.katacontainers.io

KubeCon
CloudNativeCon
North America 2018