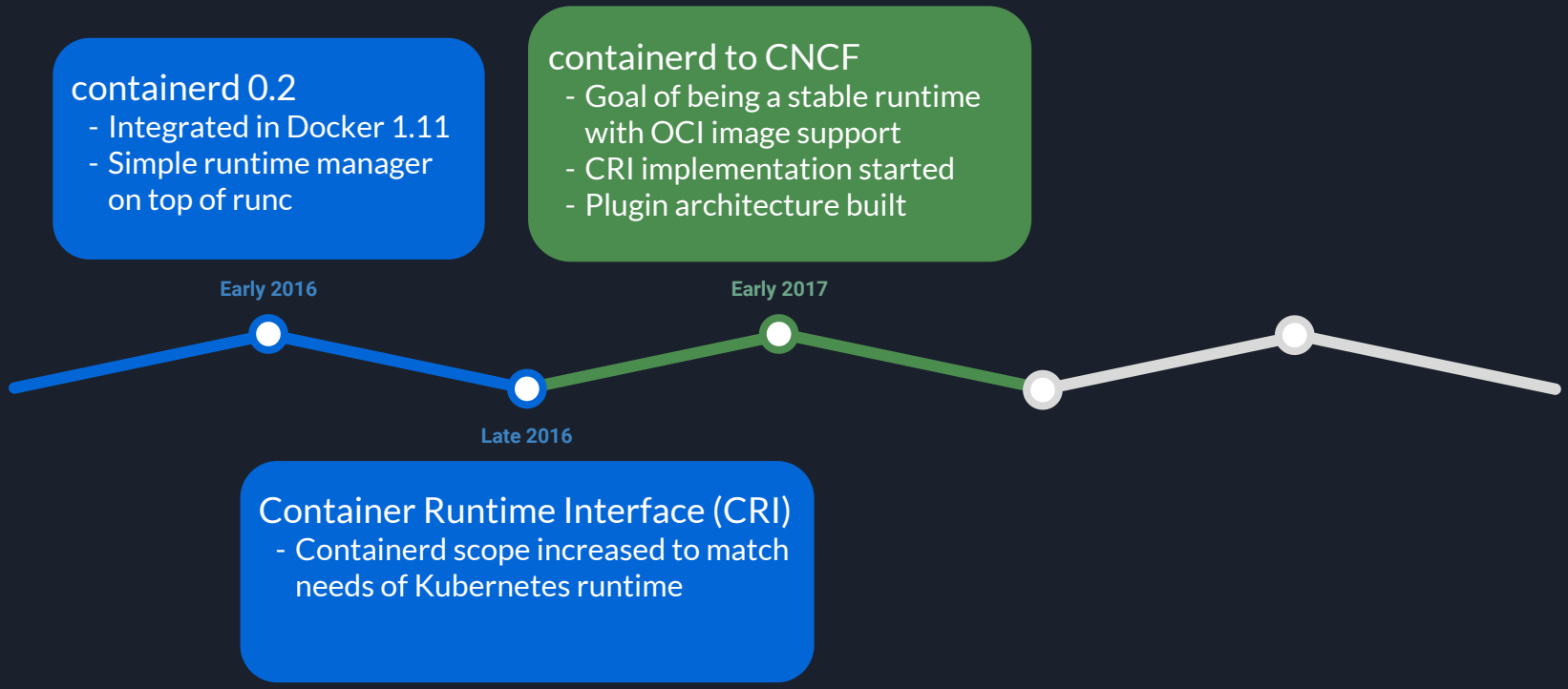
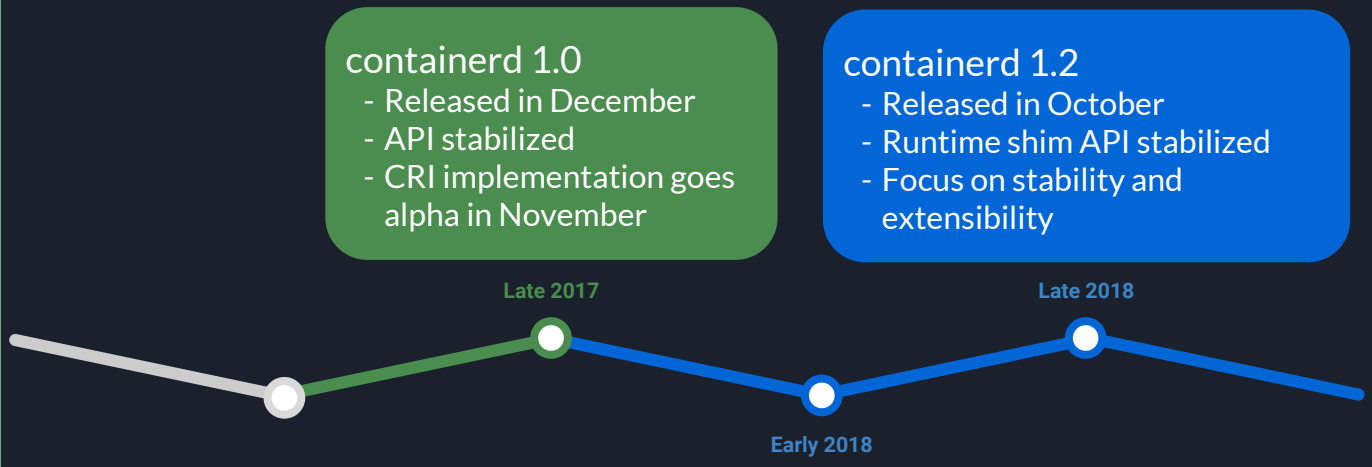
containerd intro

Kubecon Shanghai 2018
Derek McGowan (Docker)
Mike Brown (IBM)

History of containerd



History of containerd



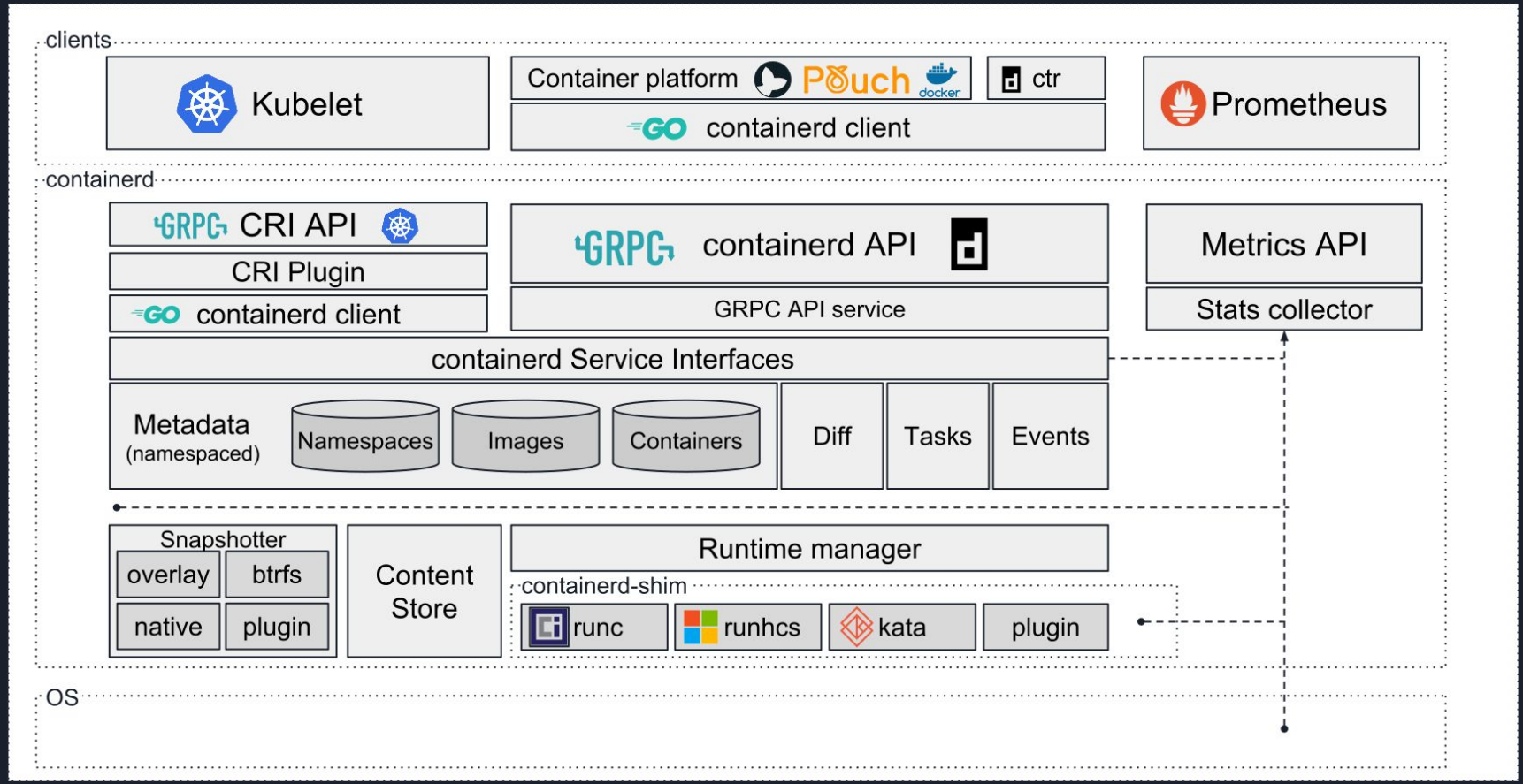
Why containerd?

- Docker
  - Scope has increased over time
  - Resource footprint not optimal for certain use cases where 3rd parties wish to replace/replicate certain higher level features that Docker provides
- Common runtime for multiple platforms
  - Kubernetes' CRI interface
  - Pouch
  - Garden
- Cloud provider integration
  - Less opinionated, more extensible for specialized cloud environments
- Owned by CNCF
- Support for OCI runtime and image specifications

Design Goals

- Loosely coupled components
  - Use any component on its own or all together
- Stable GRPC interface
- Extensibility
  - Use any runtime
  - Support any custom requirements
- Unopinionated
  - All defaults can be overwritten
  - Plugins define their configuration
- Smart client
  - Bring together decoupled components into usable toolset

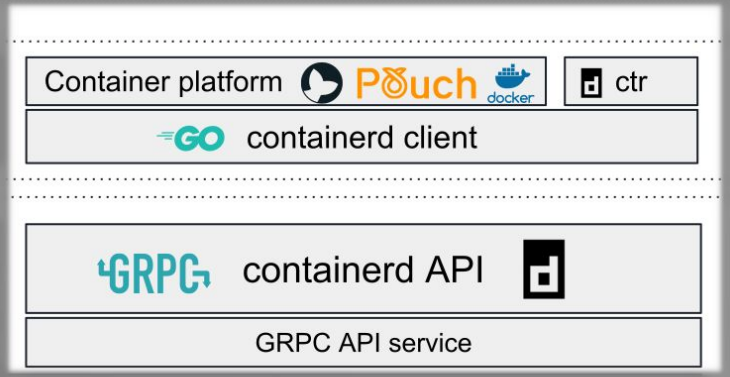
Architecture Overview



Smart Client Design

Smart client
- Higher level interface provided by client library
- Responsible for push and pull
- Direct access to low level resources (e.g. snapshots)
- Creates container OCI configuration

GRPC API

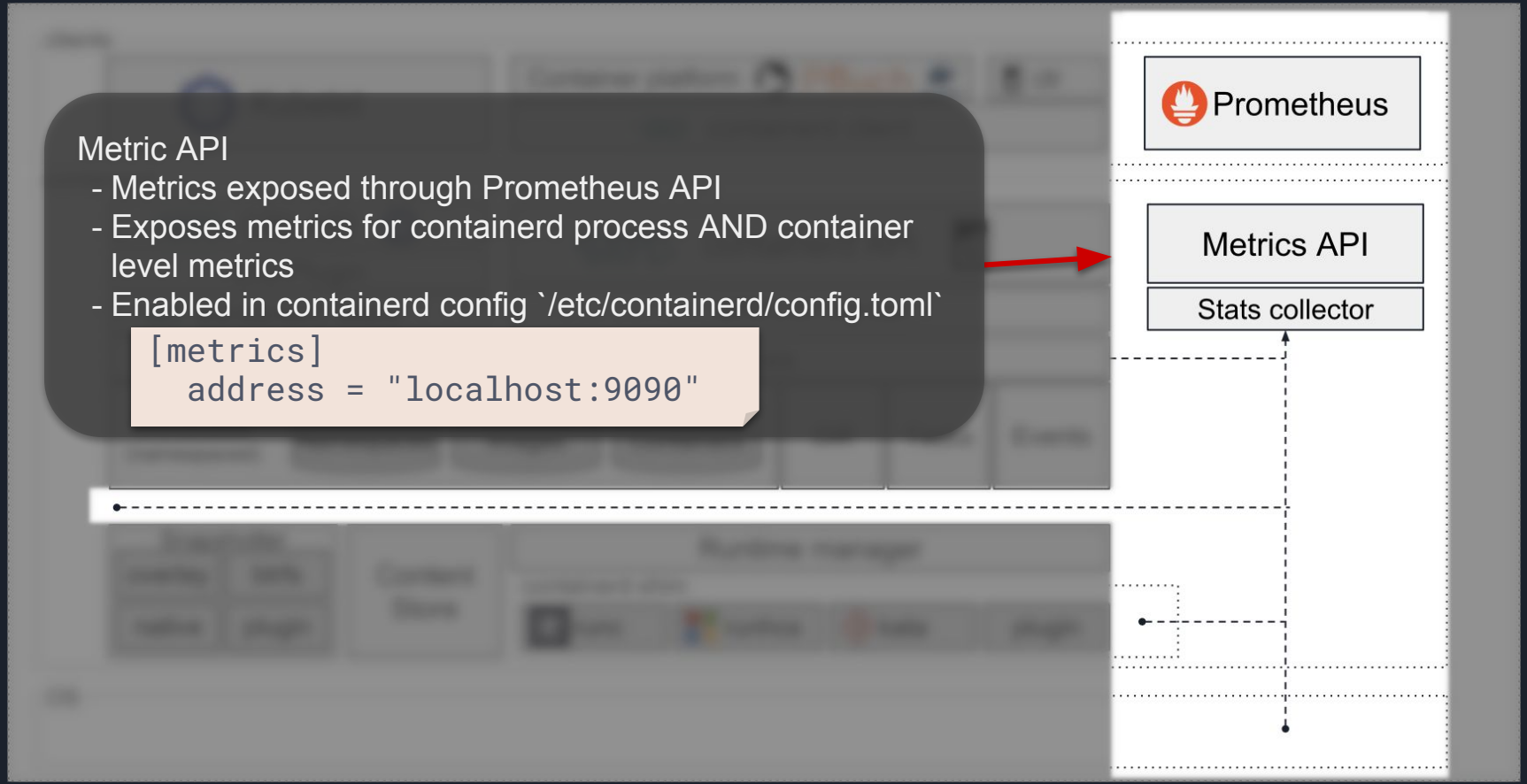


GRPC API
- Low level access to components
- Mirrors internal component interfaces
- Snapshots, Content, Containers, Task, Events, etc

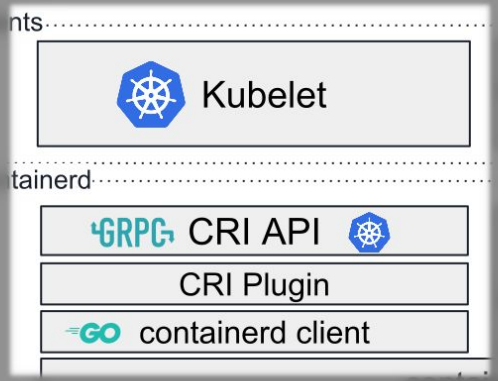
Metrics

Metric API
- Metrics exposed through Prometheus API
- Exposes metrics for containerd process AND container level metrics
- Enabled in containerd config `/etc/containerd/config.toml`

```
[metrics]
  address = "localhost:9090"
```

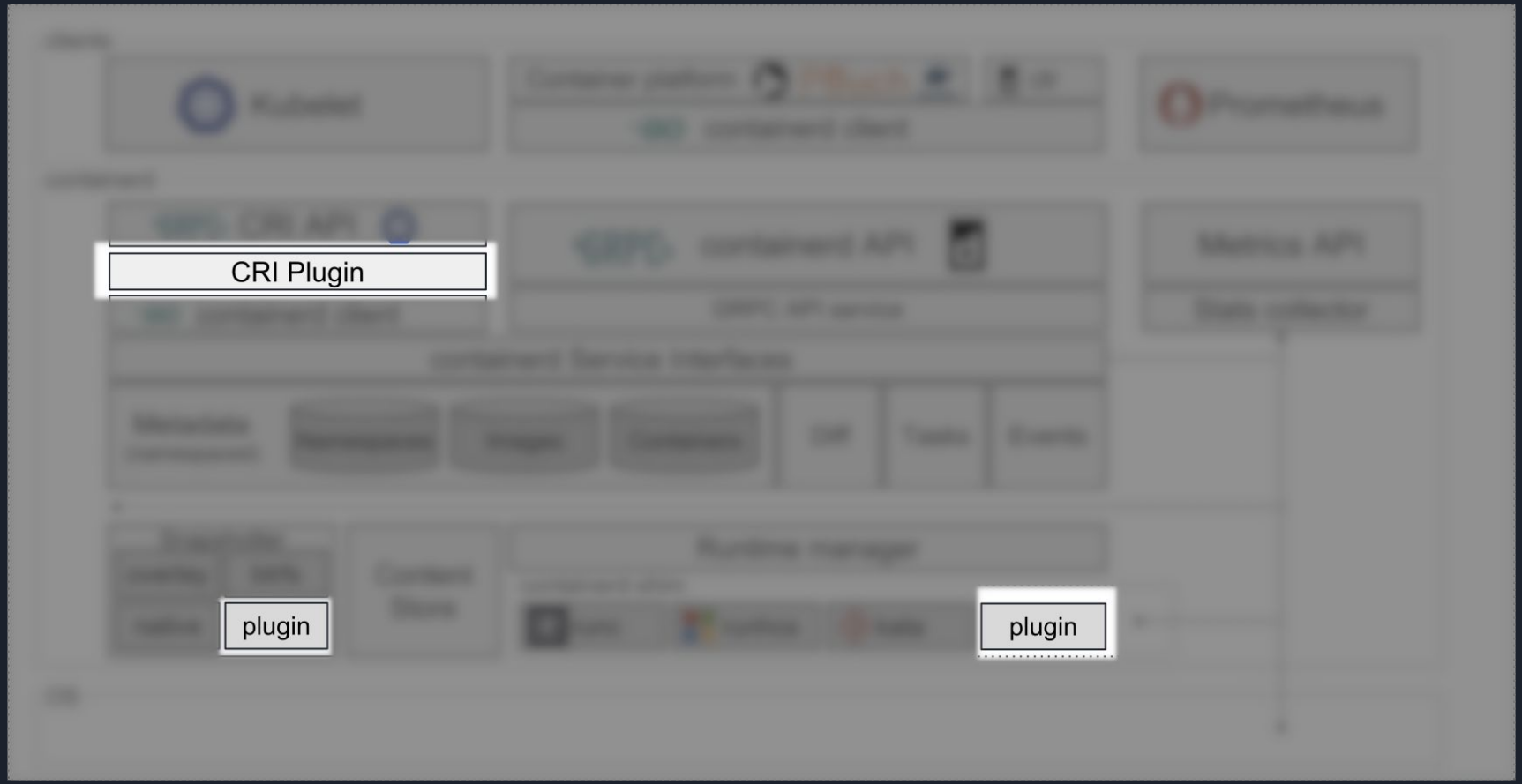


Kubernetes Support

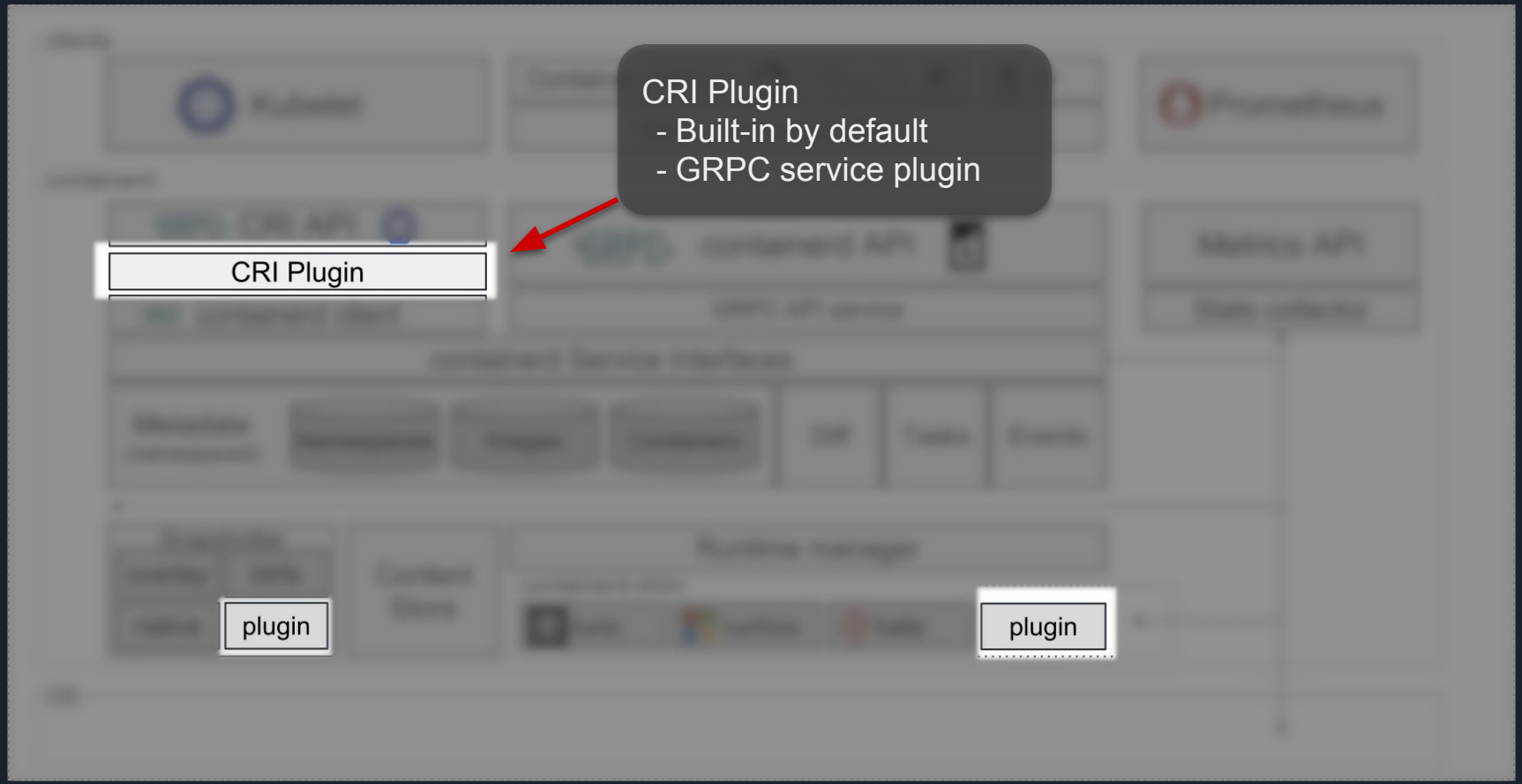


Kubernetes Runtime Support
- CRI GRPC API exposed from containerd
- Kubelet can be configured to use containerd as runtime

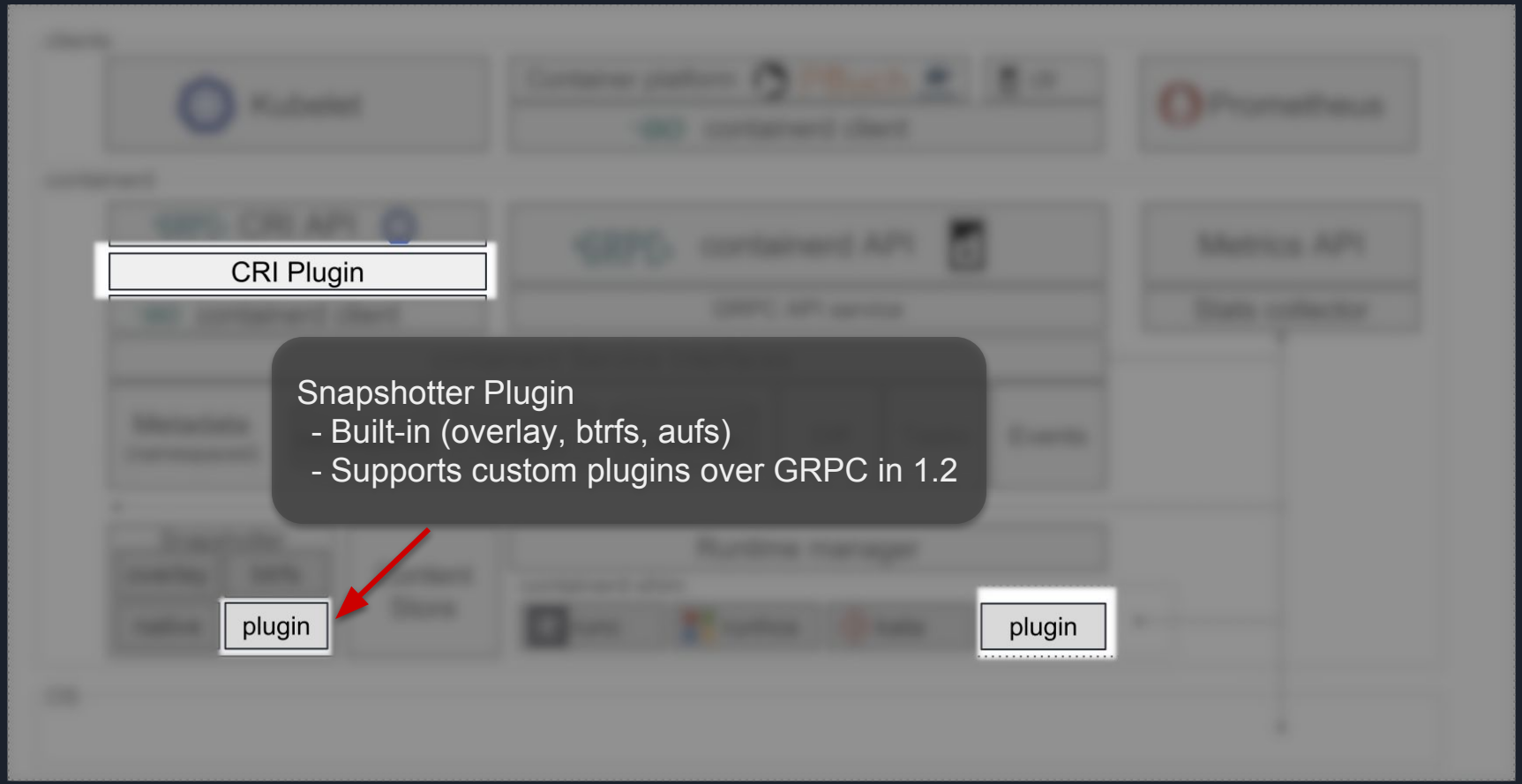
Plugins



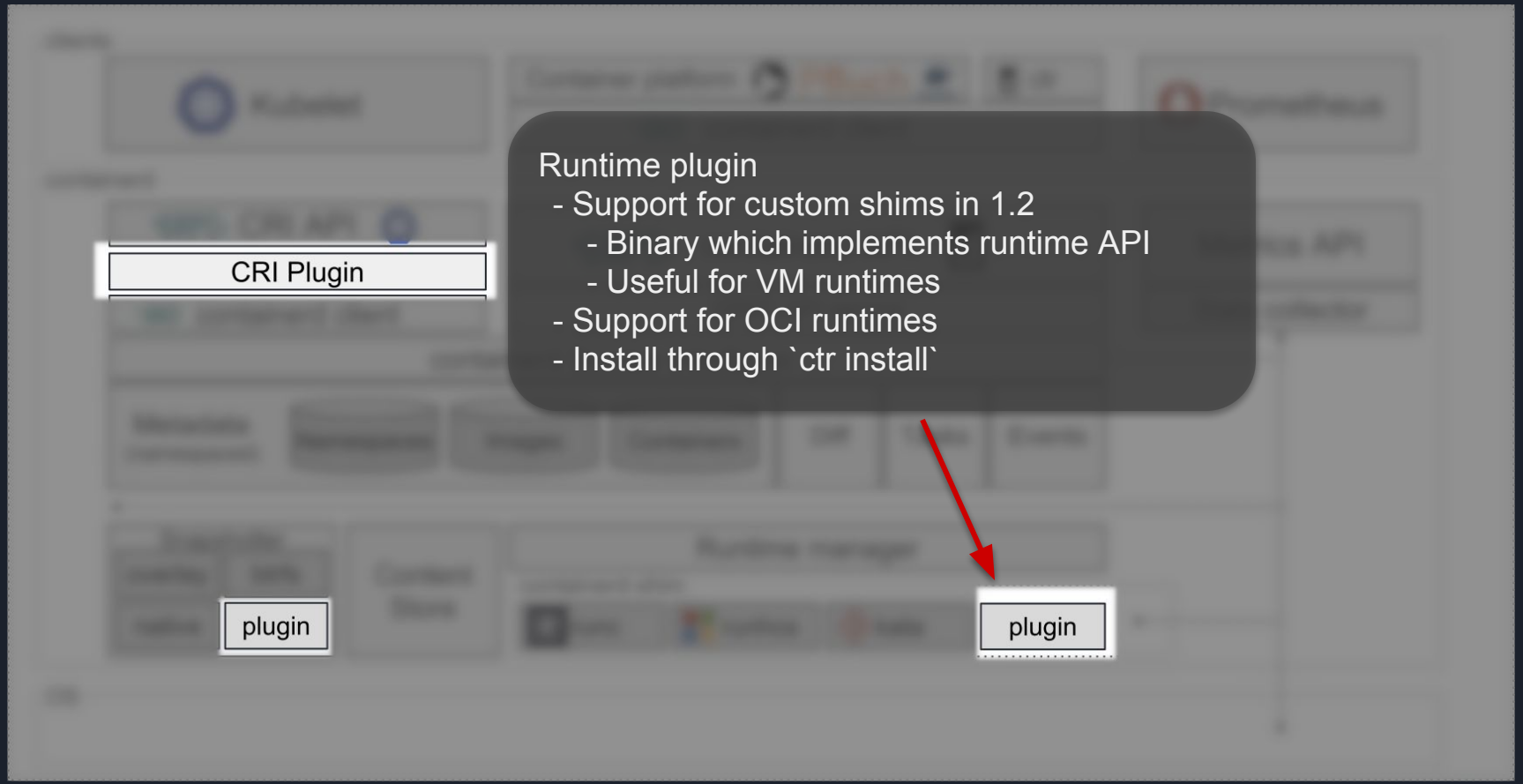
Plugins (CRI)


CRI Plugin
- Built-in by default
- GRPC service plugin
CRI Plugin
plugin
plugin

Plugins (Snapshotter)



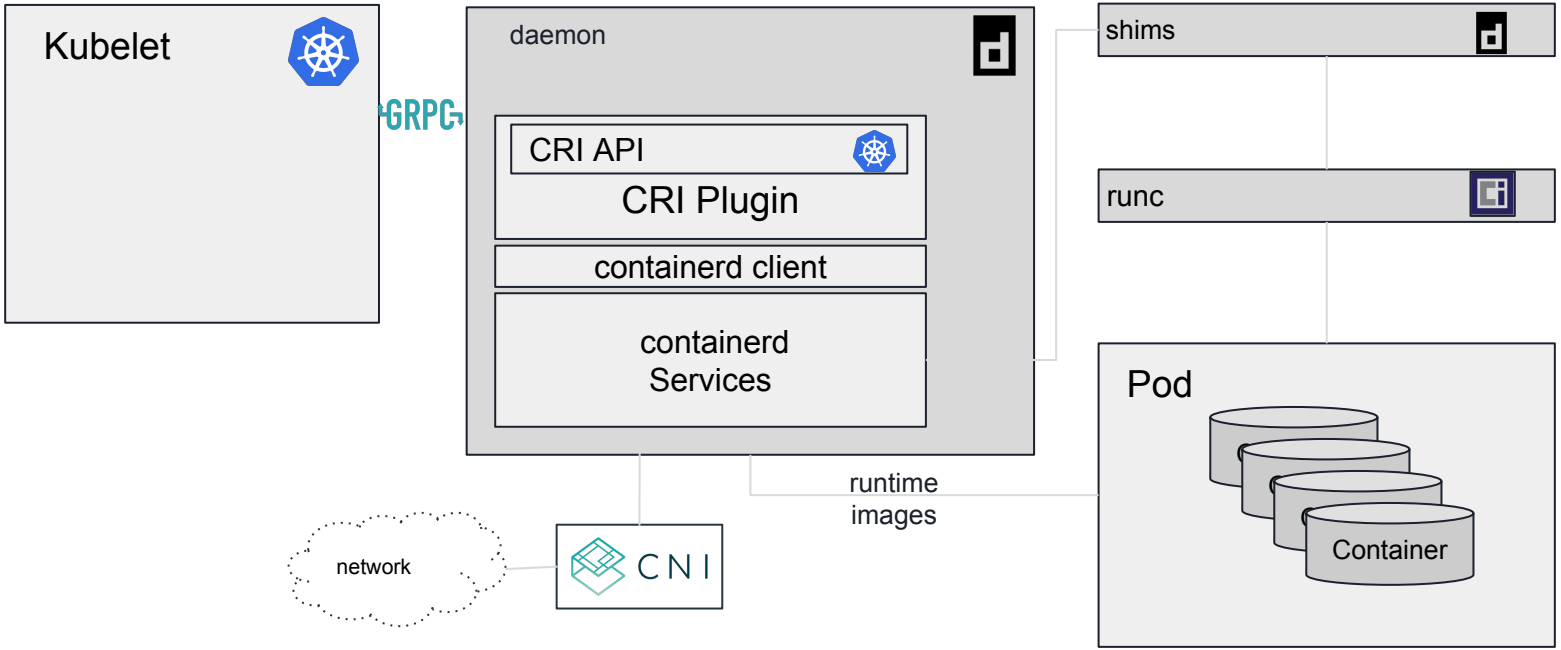
Plugins (Runtime)



More Extensibility

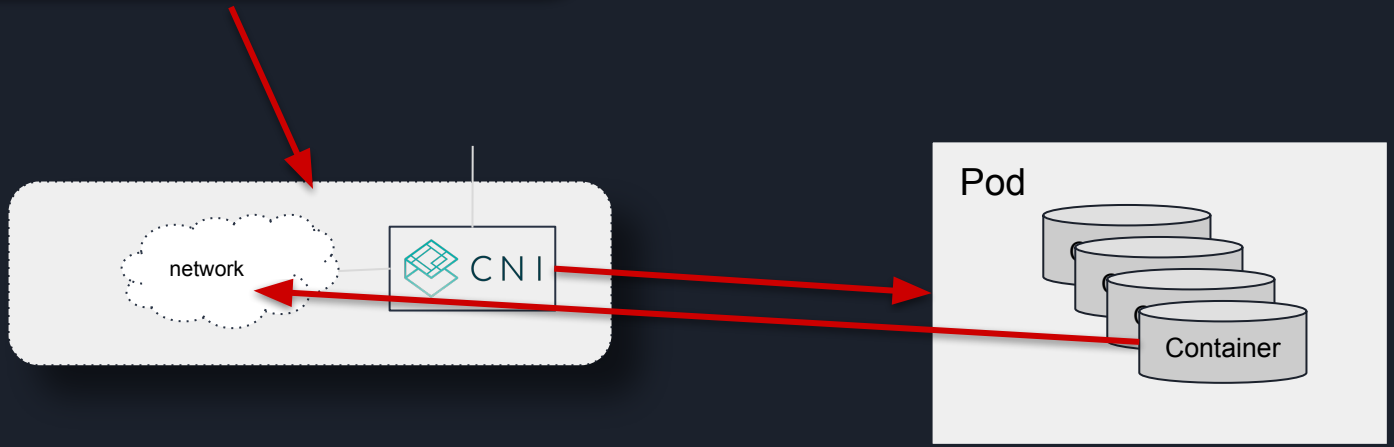
- Smart client model (Golang)
  - Resolver interface allows custom pull flow
  - Direct access to containerd interfaces
- Server plugin architecture
  - All services are self registered
  - Custom GRPC services
    - CRI is a GRPC plugin
  - Direct access to internal services

Architecture - Intro to containerd CRI



CRI plugin - (Networking)

All Networking is handled by CNI
- Support for all CNI plugins
- CRI plugin creates a network namespace for the pod via CNI
- CNI config(s) configure the CNI plugins which are used to apply the desired networking features for the pod...



Container Runtime Interface - CRI

- PodSandbox
  - Run, Stop, Remove, List, PortForward (via stream), and Status
- Containers
  - Create, Start, Stop, Remove, List, Status, Update (config), and Stats
  - ExecSync - run a command in a container, return response
  - Exec - run a command in a container asynchronously and stream the io
  - Attach - to a container - returns a stream to io of a running container
- Images
  - List, Status, Pull, Remove, FsInfo (file system information, bytes used...)
- Runtime
  - Version, Config, Status… of the Container Runtime

More: https://godoc.org/k8s.io/kubernetes/pkg/kubelet/apis/cri/runtime/v1alpha2

```
INFO[2018-10-03T20:03:38.913423245-05:00] starting containerd           revision= version=
INFO[2018-10-03T20:03:38.914989294-05:00] serving...                    address="/run/containerd/containerd.sock"
```

Tools

- ctr
  - Development tool ships with containerd, unstable (commands may change)
  - Lower level commands (directly managing snapshots, images, containers)

- crictl
  - CLI for any CRI runtime, more stable (commands less likely to change)
  - Higher level operations (pull, run, pod management)

```
INFO[2018-10-03T20:03:38.913423245-05:00] starting containerd           revision= version=
INFO[2018-10-03T20:03:38.914989294-05:00] serving...                    address="/run/containerd/containerd.sock"
```

Getting Started

Installing kubeadm:

https://kubernetes.io/docs/setup/independent/install-kubeadm/

To configure Kubernetes with containerd:

https://kubernetes.io/docs/setup/cri/#containerd

https://kubernetes.io/docs/setup/independent/create-cluster-kubeadm/

```
INFO[2018-10-03T20:03:38.913423245-05:00] starting containerd                revision= version=
INFO[2018-10-03T20:03:38.914989294-05:00] serving...                         address="/run/containerd/containerd.sock"
```

Demo

- Bring up kubernetes cluster on containred with kubeadm
- Contrast kubectl with crictl - just a bit
- Use crictl to inspect the parts of a default cluster
- Show stats with crictl
- Do a kube 101 nginx example
- Run a browser..
- Find the IP of your nginx server with crictl inspectp
- Load the page..
- Use crictl to show the nginx container log
- Bring the cluster down
- Maybe show the pods are gone but containerd is still up via crictl pods

```
INFO[2018-10-03T20:03:38.913423245-05:00] starting containerd                revision= version=
INFO[2018-10-03T20:03:38.914989294-05:00] serving...                         address="/run/containerd/containerd.sock"
```

containerd in the Cloud(s)

- Kelsey Hightower's "Kubernetes the Hard Way" deploys containerd as the kubelet runtime
- GKE beta: containerd-based K8s clusters
- IBM Cloud: containerd-based clusters for 1.11+
- Azure: OSS acs-engine includes containerd; AKS moving to containerd
- Amazon: still reviewing runtime options for EKS
- CloudFoundry: moving to containerd from runc

Build on a solid foundation
Build on containerd