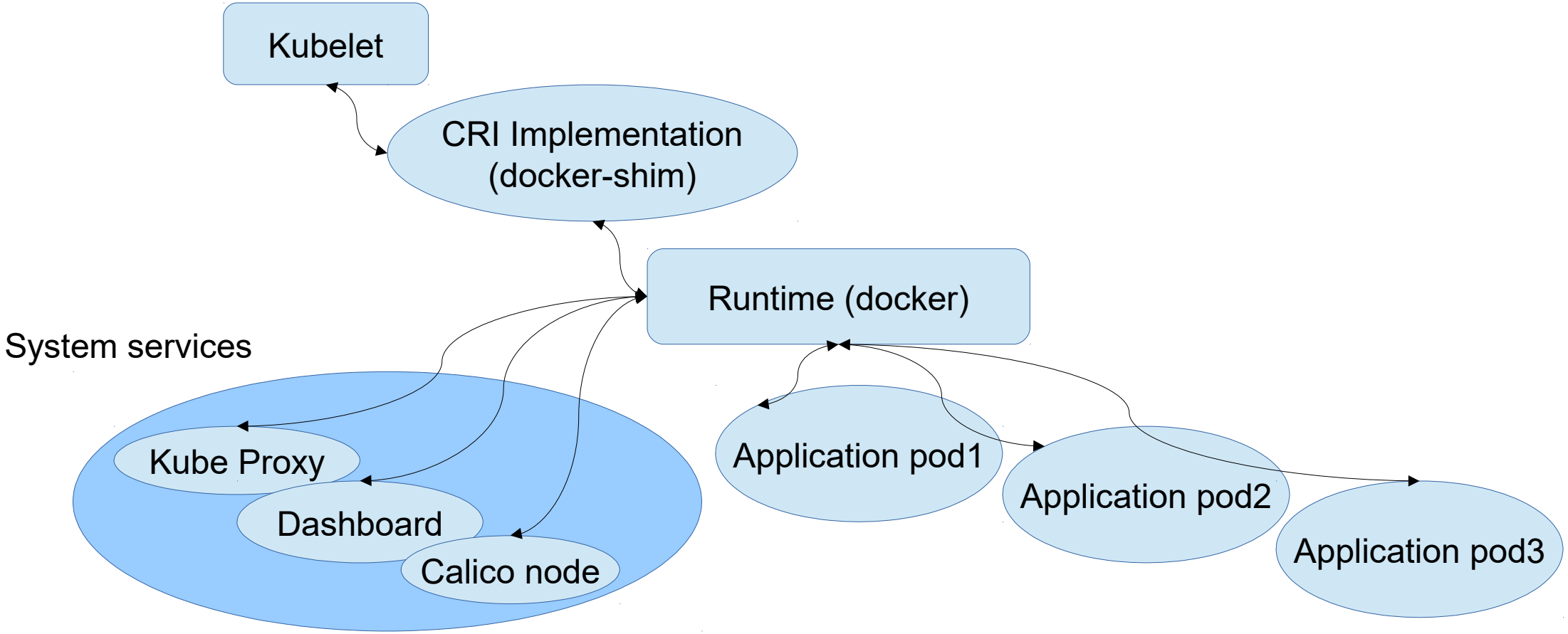
KubeCon | CloudNativeCon

North America 2017

CRI Proxy: Solving the Chicken-and-Egg Problem of Running a CRI Implementation as a DaemonSet

Piotr Skamruk, Senior Kubernetes Engineer, *Mirantis*

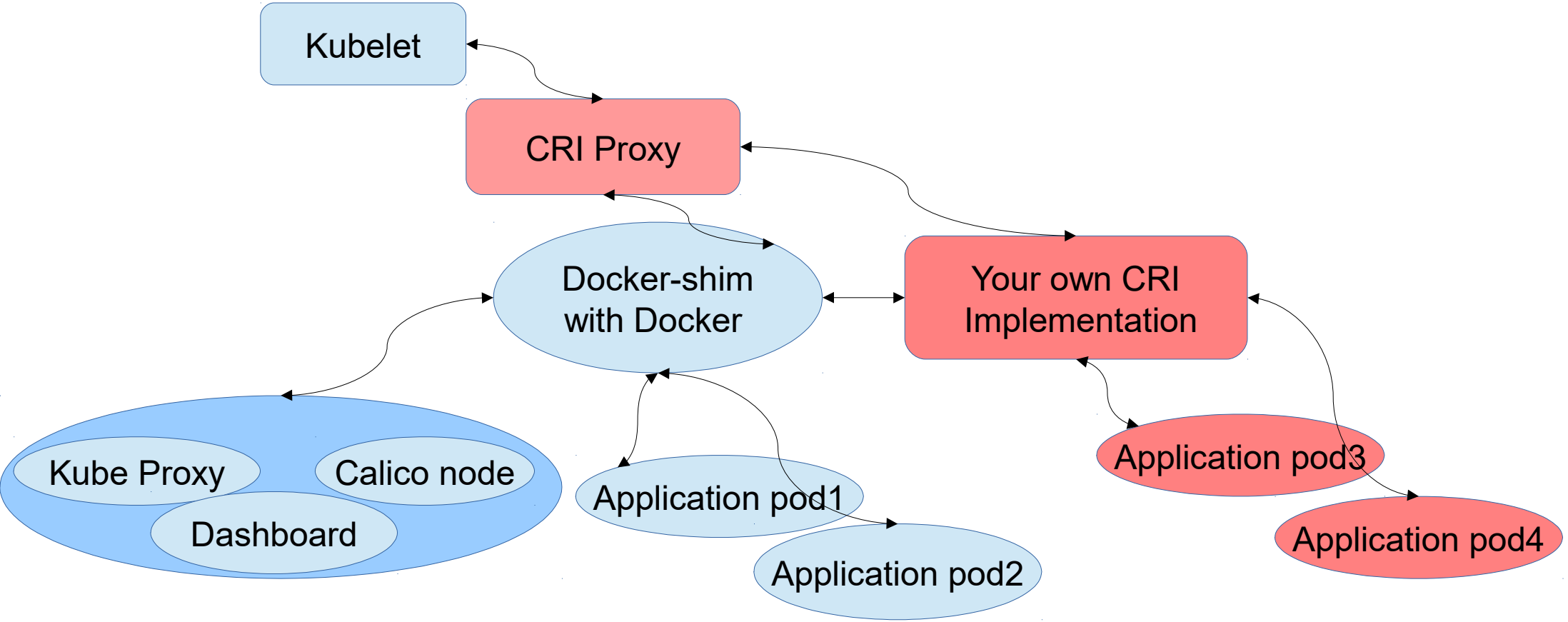
What do we have on a worker node?



Deploying a special-purpose CRI implementation

1) As example of such CRI: Virtlet
2) Special-purpose CRI may not be able to run pods such as kube-proxy, thus specialized node config is required
3) CRI implementation itself is deprived of Kubernetes deployment power as it can't run in a pod
4) It's possible to make CRI implementation itself delegate handling of some pods to docker-shim, but this only fixes 1 not 2

Let’s look on our new friend



And the bonus is...

```
00:04:32 kube-node-1 criproxy.sh[313]: I1120 00:04:32.629279     313 proxy.go:101] ENTER: /runtime.RuntimeService/ListPodSandbox(): &runtime.Li
00:04:32 kube-node-1 criproxy.sh[313]:  Filter: &runtime.PodSandboxFilter({
00:04:32 kube-node-1 criproxy.sh[313]:   Id: (string) "",
00:04:32 kube-node-1 criproxy.sh[313]:   State: &runtime.PodSandboxStateValue({
00:04:32 kube-node-1 criproxy.sh[313]:    State: (runtime.PodSandboxState) 0
00:04:32 kube-node-1 criproxy.sh[313]:   }),
00:04:32 kube-node-1 criproxy.sh[313]:   LabelSelector: (map[string]string) <nil>
00:04:32 kube-node-1 criproxy.sh[313]:  })
00:04:32 kube-node-1 criproxy.sh[313]: })
00:04:32 kube-node-1 criproxy.sh[313]: I1120 00:04:32.630550     313 proxy.go:106] LEAVE: /runtime.RuntimeService/ListPodSandbox(): &runtime.Li
00:04:32 kube-node-1 criproxy.sh[313]:  Items: ([]*runtime.PodSandbox) {
00:04:32 kube-node-1 criproxy.sh[313]:   &runtime.PodSandbox({
00:04:32 kube-node-1 criproxy.sh[313]:    Id: "f65c2109e71b6d0aa4cf45472fcff9360d01dce3f960051de21b622f33ba9a6b",
00:04:32 kube-node-1 criproxy.sh[313]:    Metadata: &runtime.PodSandboxMetadata({
00:04:32 kube-node-1 criproxy.sh[313]:     Name: "kubernetes-dashboard-1136505239-qldd2",
00:04:32 kube-node-1 criproxy.sh[313]:     Uid: "38875a67-cb9b-11e7-bcec-024256428e7a",
00:04:32 kube-node-1 criproxy.sh[313]:     Namespace: "kube-system",
00:04:32 kube-node-1 criproxy.sh[313]:     Attempt: (uint32) 0
00:04:32 kube-node-1 criproxy.sh[313]:    }),
00:04:32 kube-node-1 criproxy.sh[313]:    State: (runtime.PodSandboxState) 0,
00:04:32 kube-node-1 criproxy.sh[313]:    CreatedAt: (int64) 1510925319000000000,
00:04:32 kube-node-1 criproxy.sh[313]:    Labels: (map[string]string) {
00:04:32 kube-node-1 criproxy.sh[313]:     "app": "kubernetes-dashboard",
00:04:32 kube-node-1 criproxy.sh[313]:     "io.kubernetes.pod.name": "kubernetes-dashboard-1136505239-qldd2",
00:04:32 kube-node-1 criproxy.sh[313]:     "io.kubernetes.pod.namespace": "kube-system",
00:04:32 kube-node-1 criproxy.sh[313]:     "io.kubernetes.pod.uid": "38875a67-cb9b-11e7-bcec-024256428e7a",
00:04:32 kube-node-1 criproxy.sh[313]:     "pod-template-hash": "1136505239"
00:04:32 kube-node-1 criproxy.sh[313]:    },
00:04:32 kube-node-1 criproxy.sh[313]:    Annotations: (map[string]string) {
00:04:32 kube-node-1 criproxy.sh[313]:     "kubernetes.io/config.seen": "2017-11-17T13:28:38.699893615Z",
00:04:32 kube-node-1 criproxy.sh[313]:     "kubernetes.io/created-by": "{\"kind\":\"SerializedReference\",\"apiVersion\":\"v1\",\"reference\":{
00:04:32 kube-node-1 criproxy.sh[313]:     "scheduler.alpha.kubernetes.io/tolerations": "[\n  {\n    \"key\": \"dedicated\",\n    \"operator\":
00:04:32 kube-node-1 criproxy.sh[313]:     "kubernetes.io/config.source": "api"
00:04:32 kube-node-1 criproxy.sh[313]:    }
00:04:32 kube-node-1 criproxy.sh[313]:   }),
00:04:32 kube-node-1 criproxy.sh[313]:   &runtime.PodSandbox({
00:04:32 kube-node-1 criproxy.sh[313]:    Id: "217fca69226fe54e9a33a7cdeef6b9661df8706c657a771341b06810cd565d50",
```

Thank you

You can find out more about the project at: https://github.com/Mirantis/criproxy