法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，讲师及小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

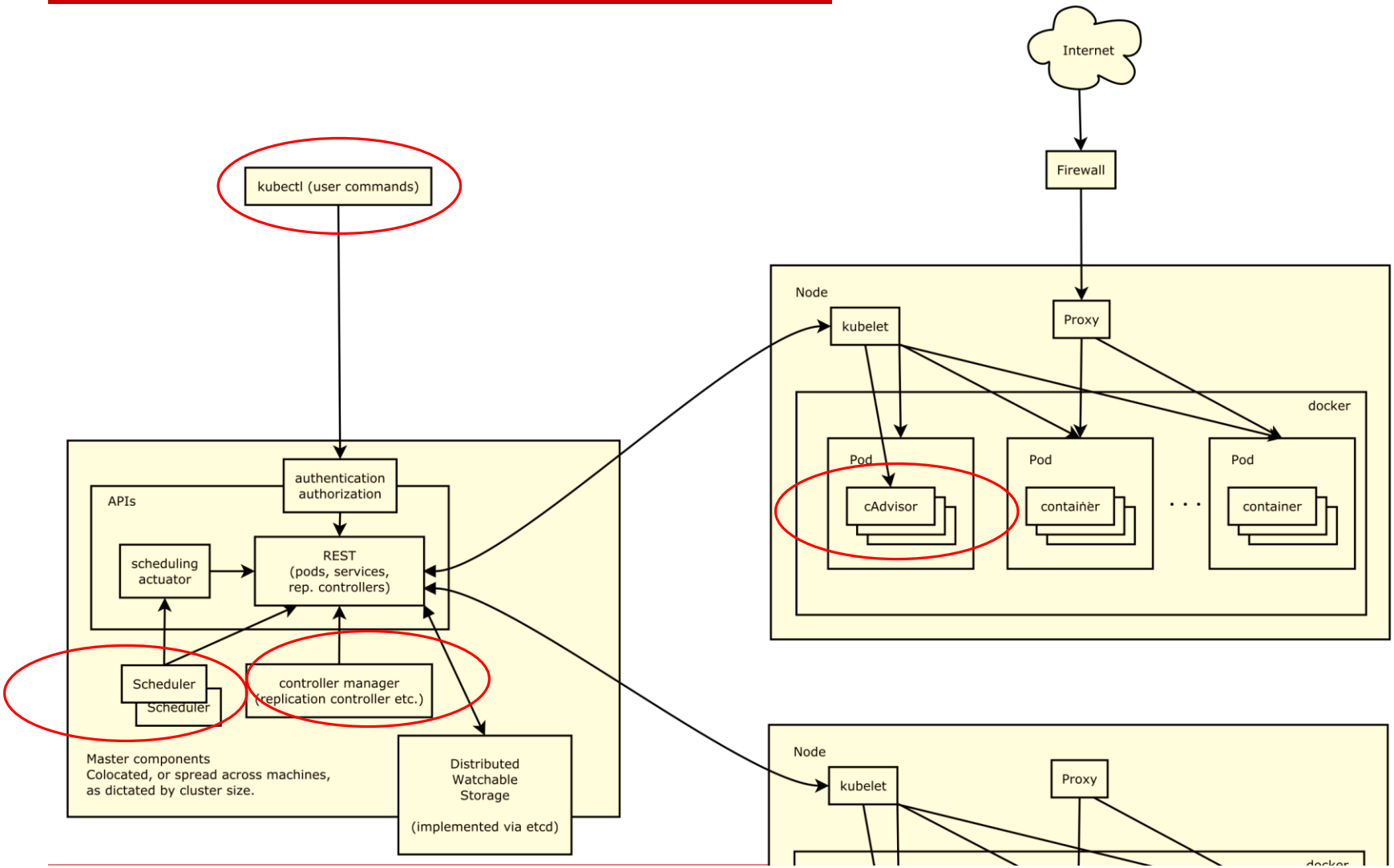
- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

Kubernetes扩展开发

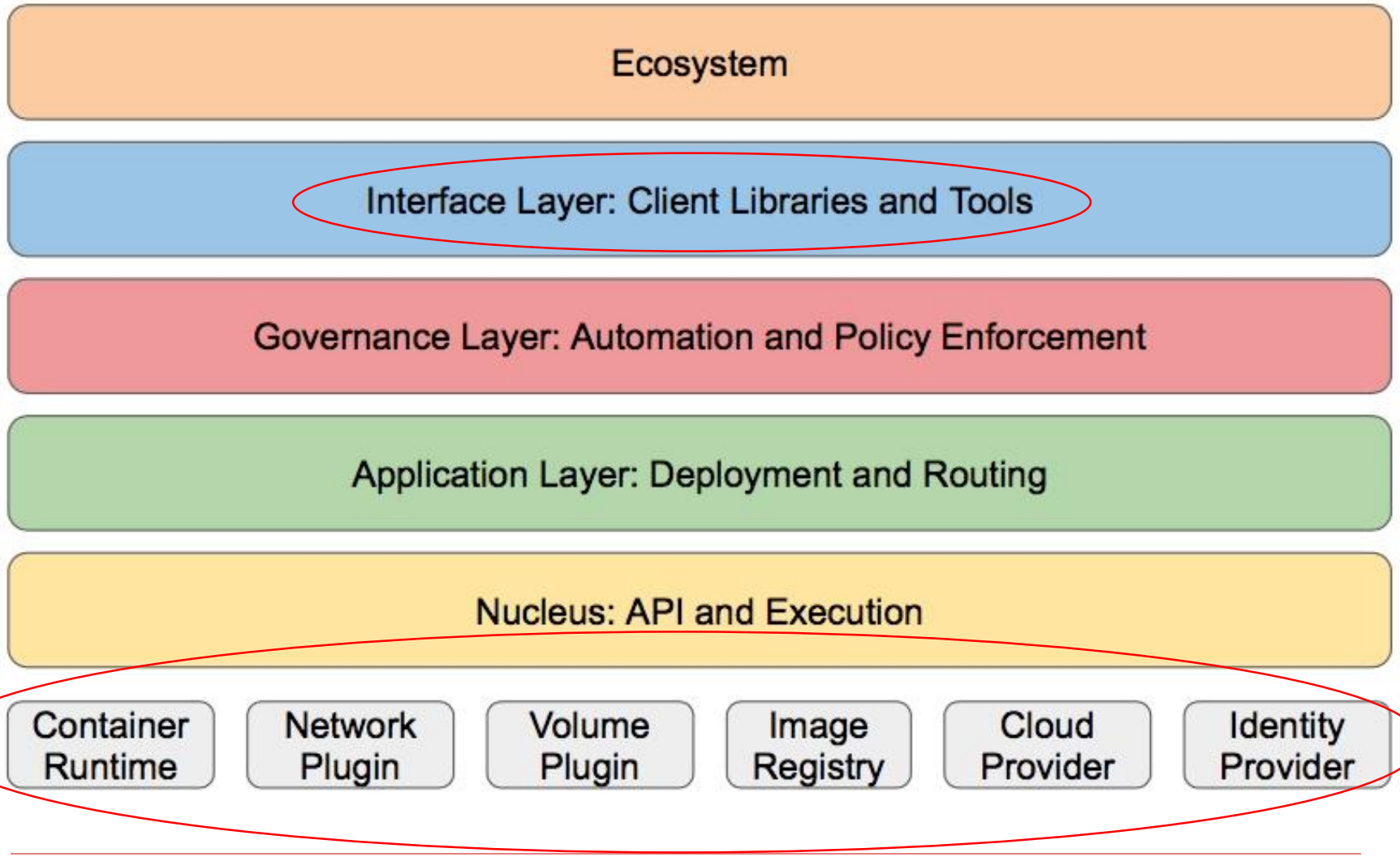
目录

1. 扩展点简介
2. k8s核心机制
3. 各扩展点入门实操
4. CRI/CNI/CSI

1. Kubernetes扩展点简介



1. Kubernetes扩展点简介



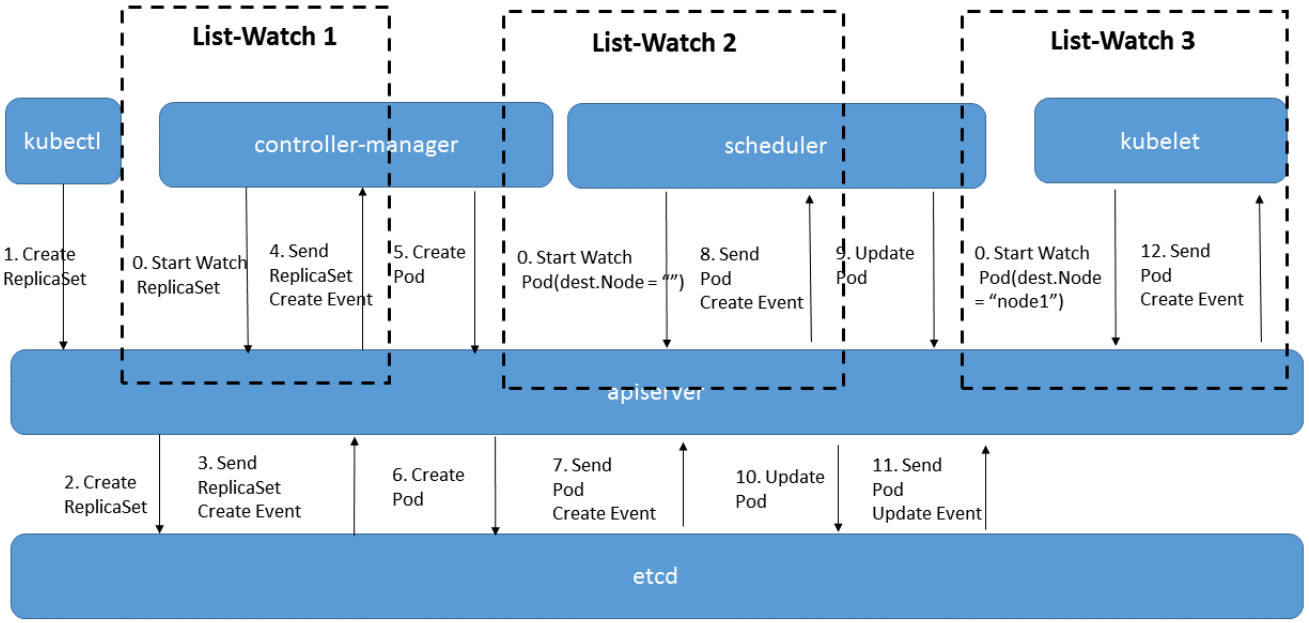
1. Kubernetes扩展点简介

- 客户端
  - Shell脚本调用kubectl
  - Go/python 调用client SDK
- 调度器
  - 彻底实现自己的调度器
  - 实现一个第三方的调度插件，通过http调用等方式插入现有调度算法
- Controller-Manager
  - CRD，Operator模式定义自己的API对象并运行为控制器

1. Kubernetes扩展点简介

- 容器运行时
  - CRI：container runtime interface
- 网络
  - CNI：container network interface
- 存储
  - CSI：container storage interface

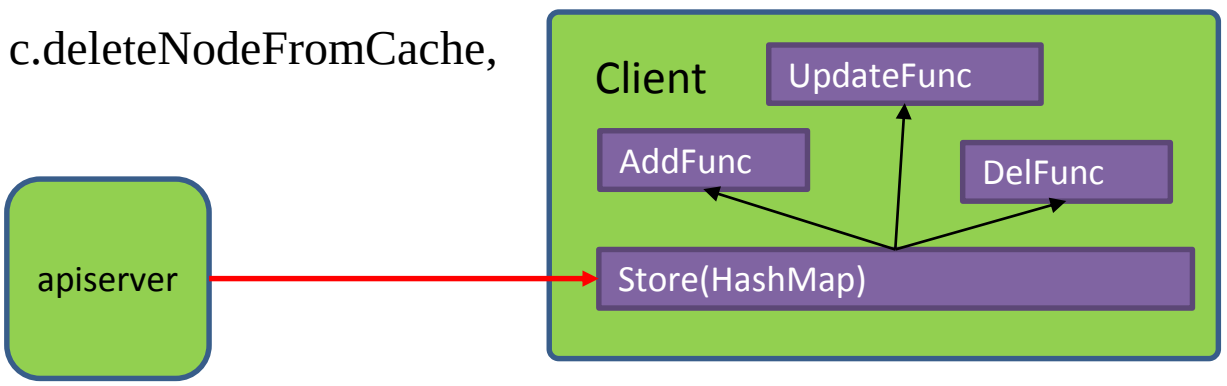
2. k8s核心机制 — list-watch



- list-watch由各个组件向apiserver发起restful http请求，本质是一种Pub-Sub过程
- list-watch是可以带条件的，客户端只关心部分数据
- list是watch失败，数据过于陈旧的弥补条件

2. k8s核心机制 — list-watch

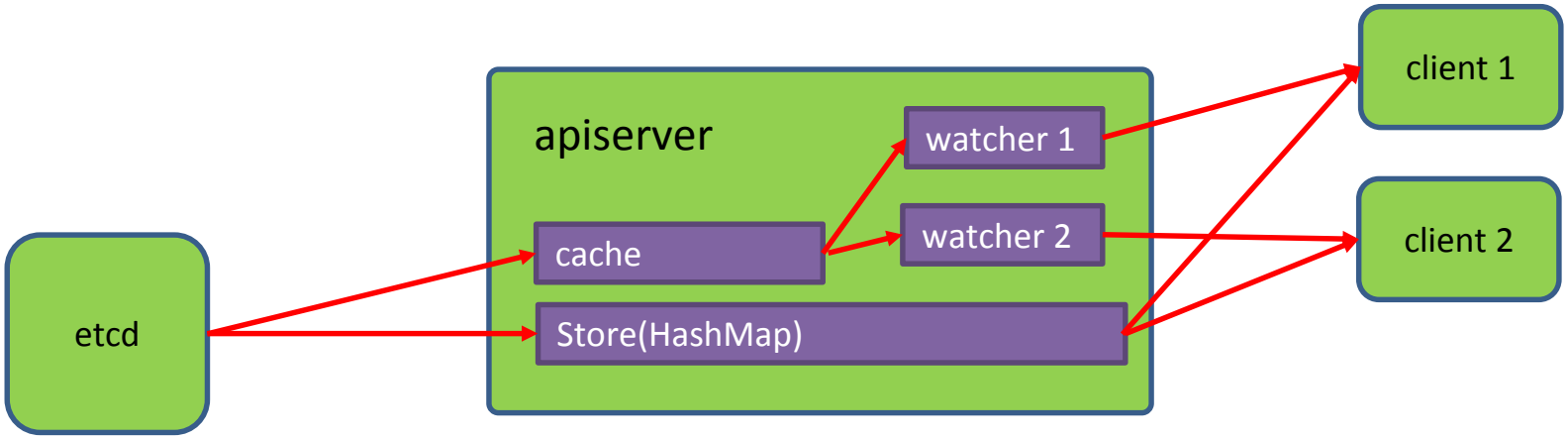
```
c.NodeLister.Store, c.nodePopulator = framework.NewInformer(
        c.createNodeLW(),
        &api.Node{},
        0,
        framework.ResourceEventHandlerFuncs{
            AddFunc:    c.addNodeToCache,
            UpdateFunc: c.updateNodeInCache,
            DeleteFunc: c.deleteNodeFromCache,
        },
)
```



2. k8s核心机制 — list-watch

```
type watchCache struct {
        // slide window
    cache      []watchCacheElement

    // all data
    store cache.Store
}
```



3. Client入门

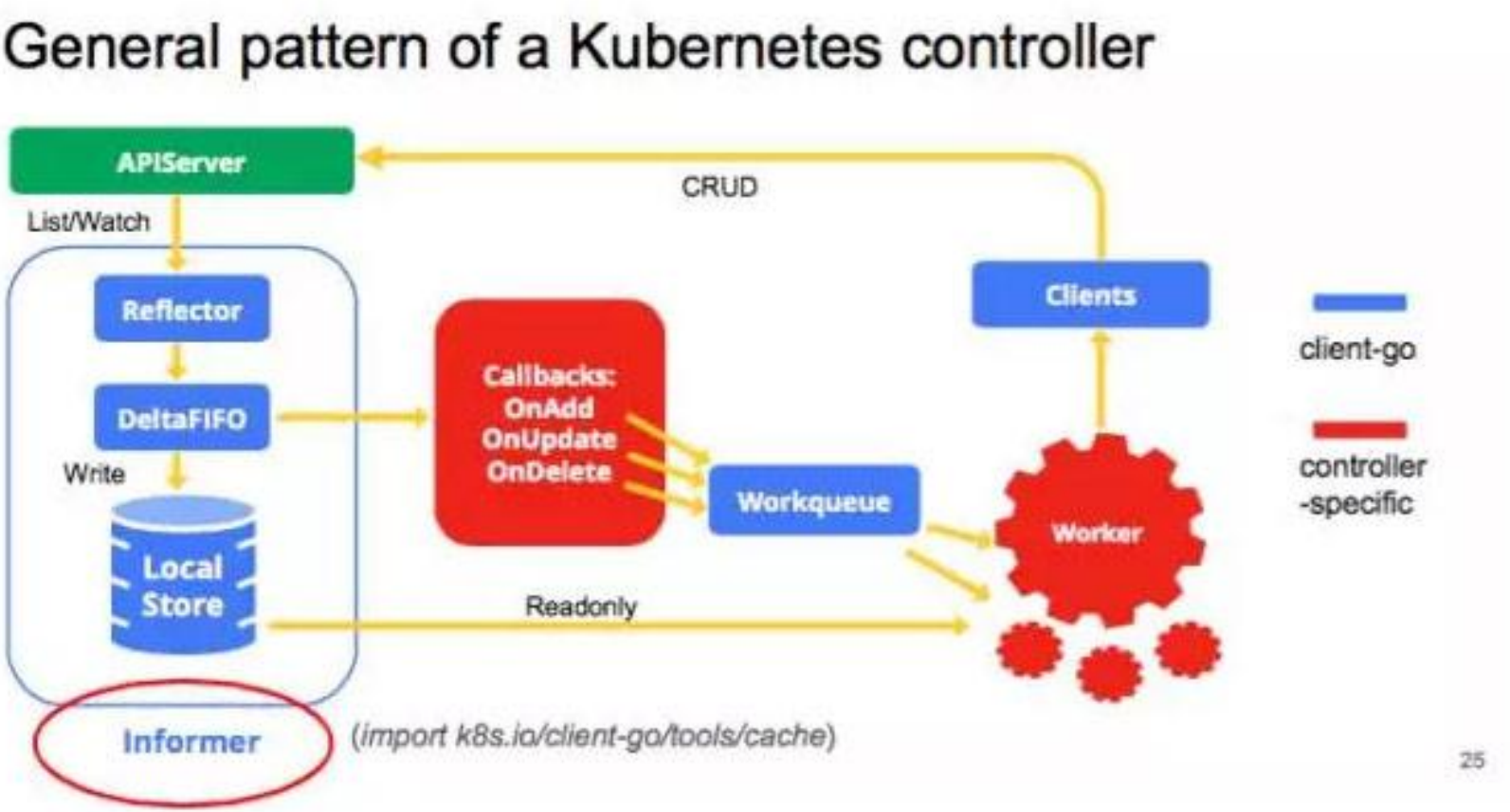
方式	特点	支持者
Kubernetes dashboard	直接通过Web UI进行操作，简单直接，可定制化程度低	官方支持
kubectl	命令行操作，功能最全，但是比较复杂，适合对其进行进一步的分装，定制功能，版本适配最好	官方支持
client-go	从kubernetes的代码中抽离出来的客户端包，简单易用，但需要小心区分kubernetes的API版本	官方支持
client-python	python客户端，kubernetes-incubator	官方支持
Java client	fabric8中的一部分，kubernetes的java客户端	redhat

3. Client入门

- Client-go 案例

https://github.com/rootsongjc/kubernetes-client-go-sample

3. 使用Client实现CRD Controller



3. 使用Client实现Operator

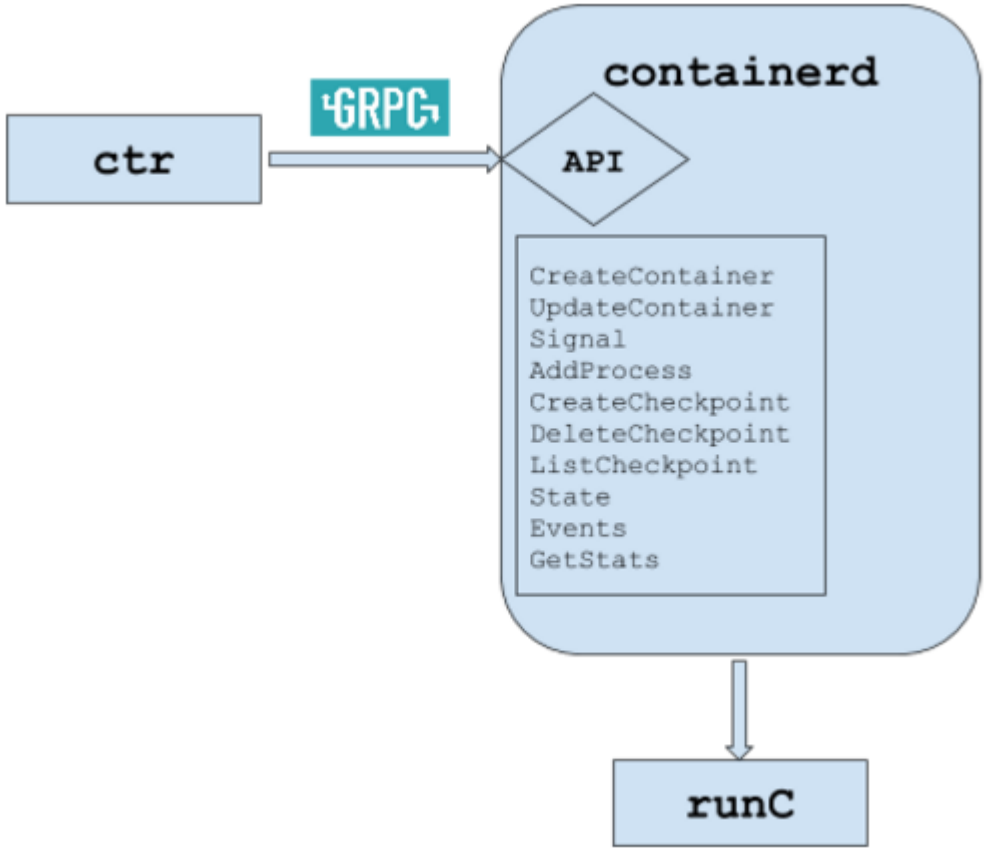
- 在单个Deployment中定义Operator，如：
  https://coreos.com/operators/etcd/latest/deployment.yaml
- 需要为Operator创建一个新的自定义类型CRD，这样用户就可以使用该对象来创建实例
- Operator应该利用Kubernetes中内建的原语，如Deployment、Service这些经过充分测试的对象，这样也便于理解
- Operator应该向后兼容，始终了解用户在之前版本中创建的资源
- 当Operator被停止或删除时，Operator创建的应用实例应该不受影响
- Operator应该让用户能够根据版本声明来选择所需版本和编排应用程序升级。不升级软件是操作错误和安全问题的常见来源，Operator可以帮助用户更加自信地解决这一问题。
- Operator应该进行“Chaos Monkey”测试，以模拟Pod、配置和网络故障的情况下的行为。

3. Operator代码分析

https://github.com/coreos/etcd-operator

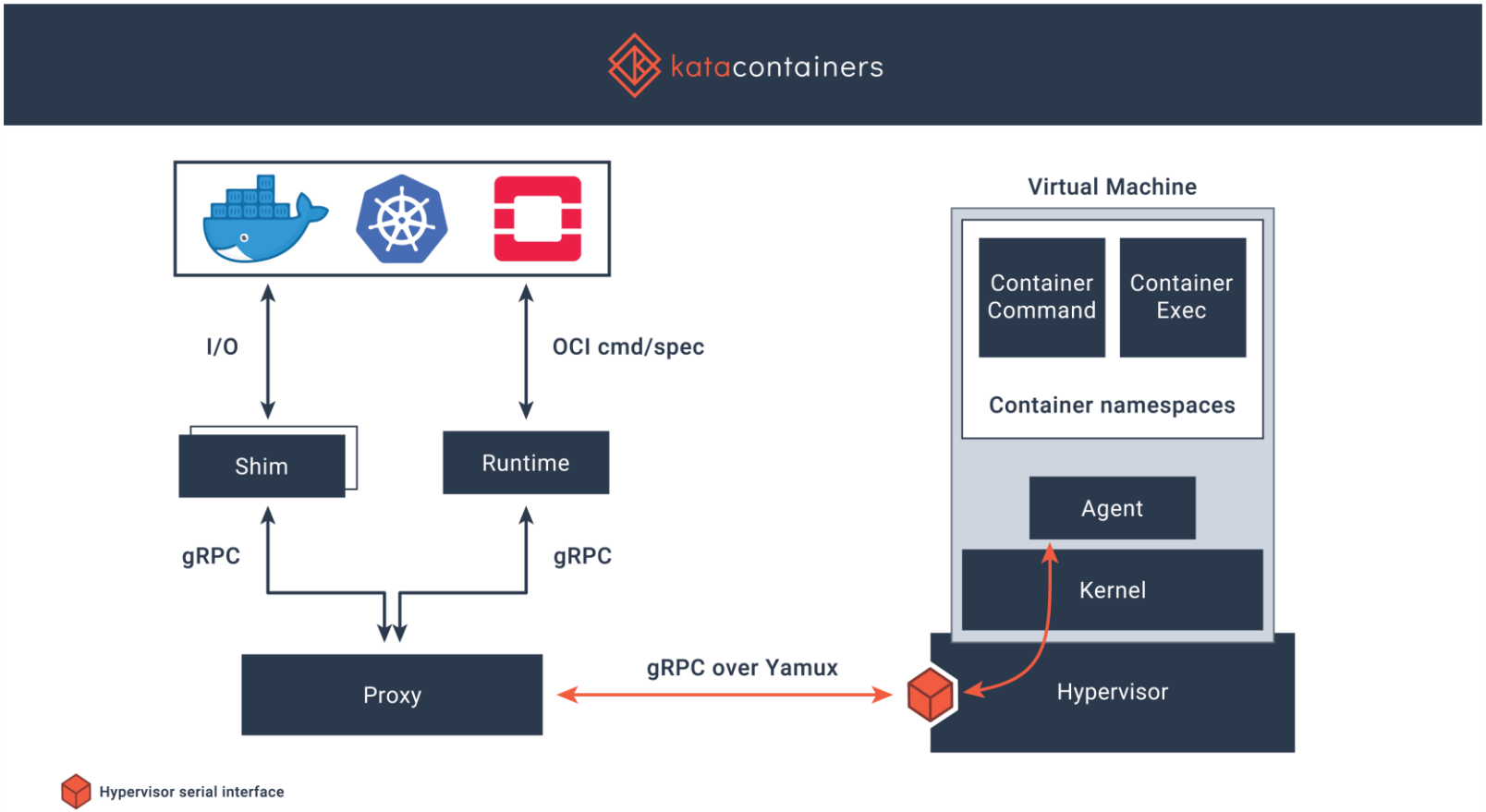
4. CRI -- runC/runV?

http://dockone.io/article/914



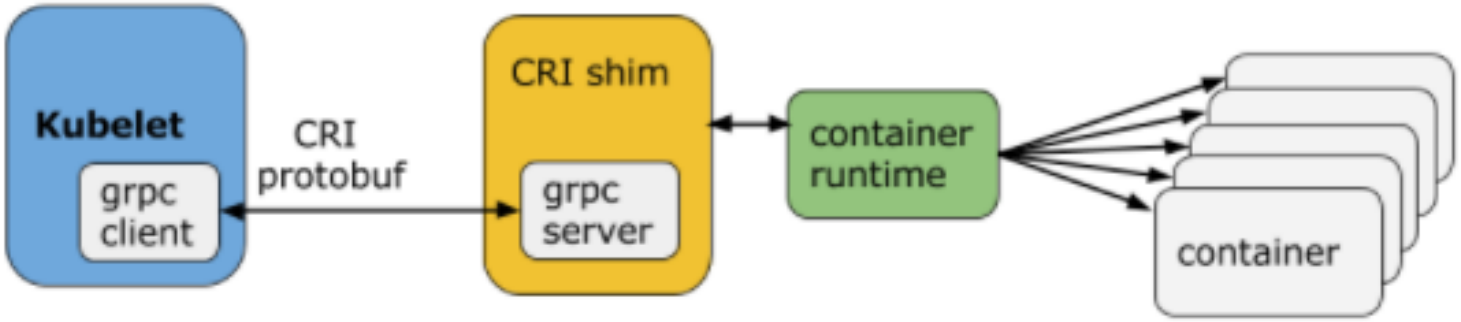
4. CRI -- runC/runV?

https://katacontainers.io/



4. CRI -- runC/runV?

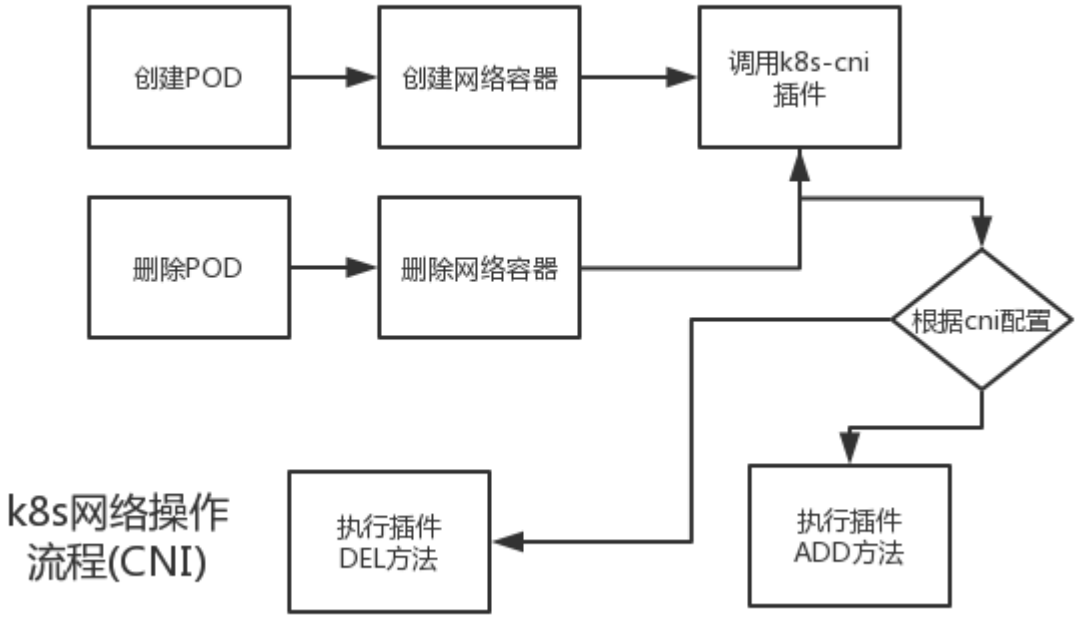
https://jimmysong.io/kubernetes-handbook/concepts/cri.html



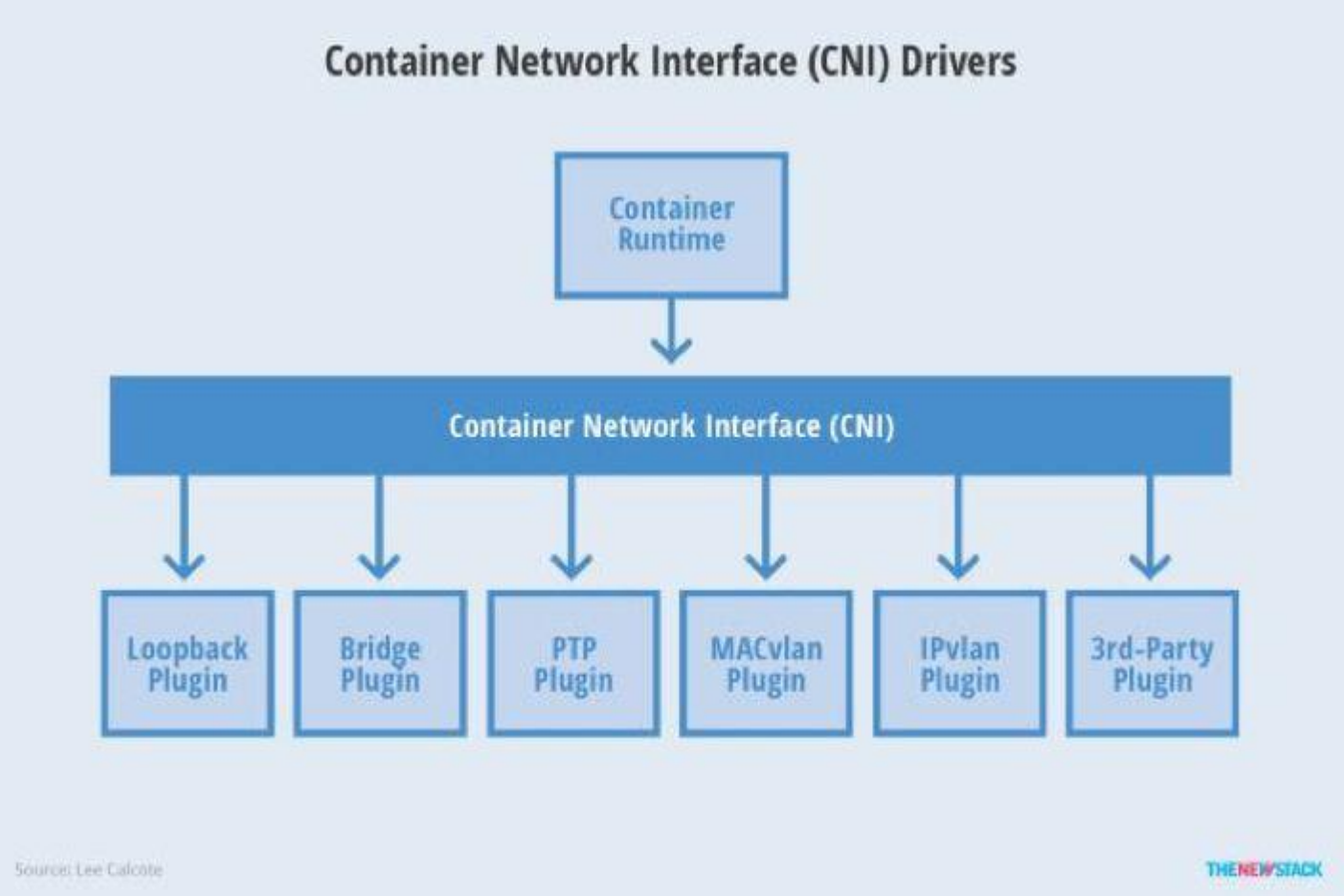
- 启动/停止/删除 容器
- 下载/删除 镜像

4. CNI

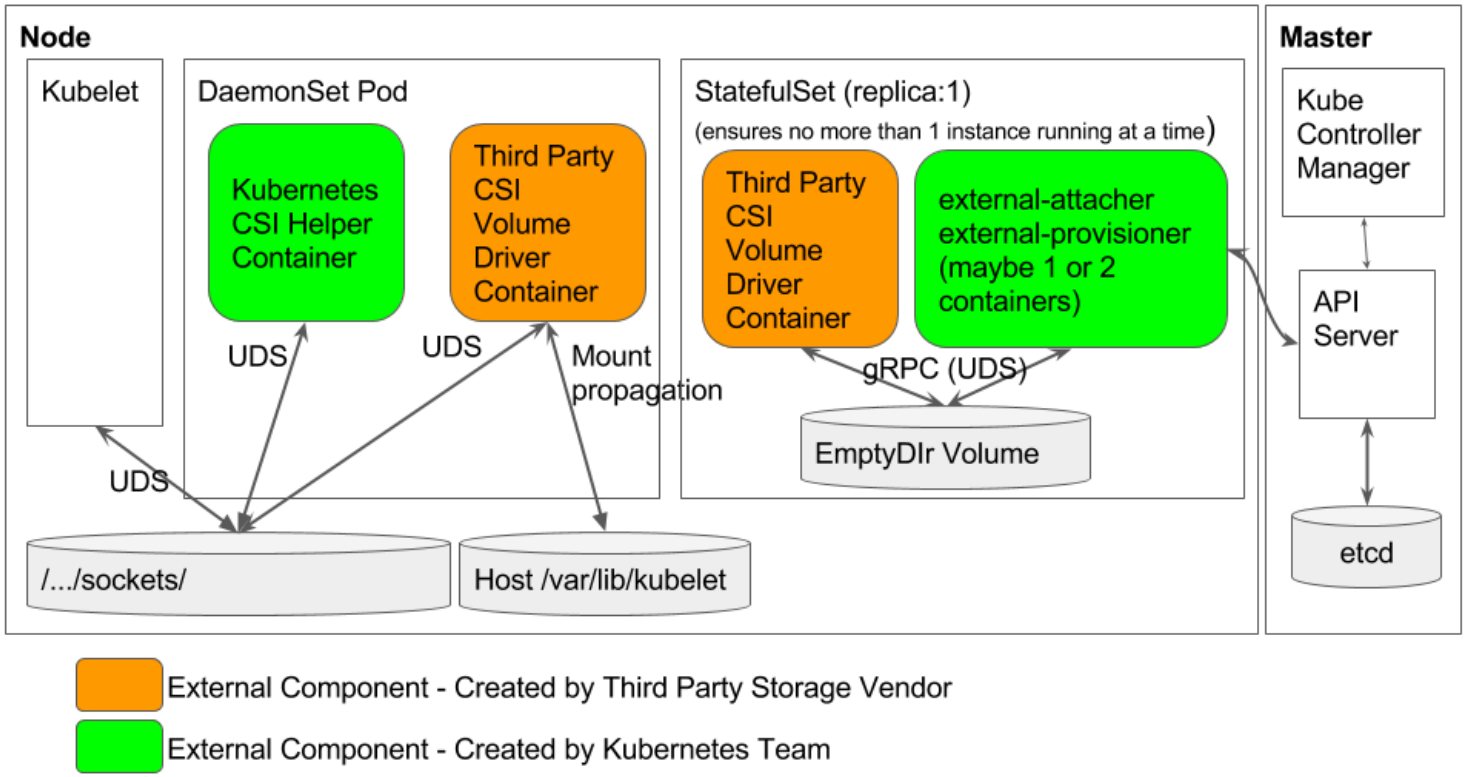
- https://github.com/containernetworking/cni/blob/master/SPEC.md



4. CNI


Container Network Interface (CNI) Drivers
Container Runtime
Container Network Interface (CNI)
Loopback Plugin
Bridge Plugin
PTP Plugin
MACvlan Plugin
IPvlan Plugin
3rd-Party Plugin
Source: Lee Calcote
THENEWSTACK

4. CSI



- [CSI Volume Plugins in Kubernetes Design Doc]
- [Kubernetes 1.9 容器存储接口（CSI）Alpha 版本全解析]

扩展阅读

《Extending your Kubernetes Cluster》

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：大数据分析挖掘
- 新浪微博：ChinaHadoop