

What Happens When I Type `kubectl run`? 🤔



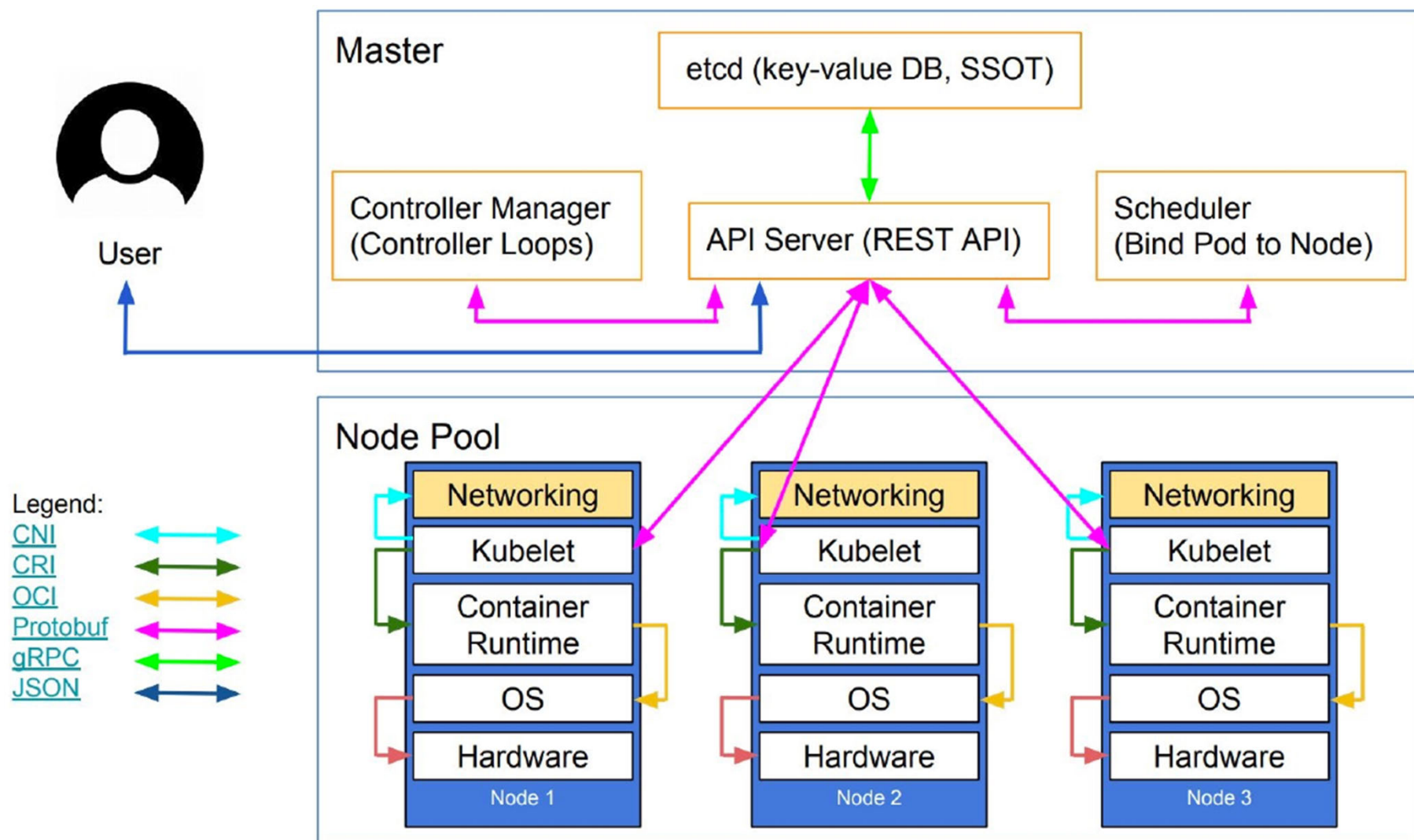
@Luminous Moonlight

Presented by **linxuyalun**

<https://github.com/jamiehannaford/what-happens-when-k8s>

2020-11-11

Kubernetes' high-level component architecture

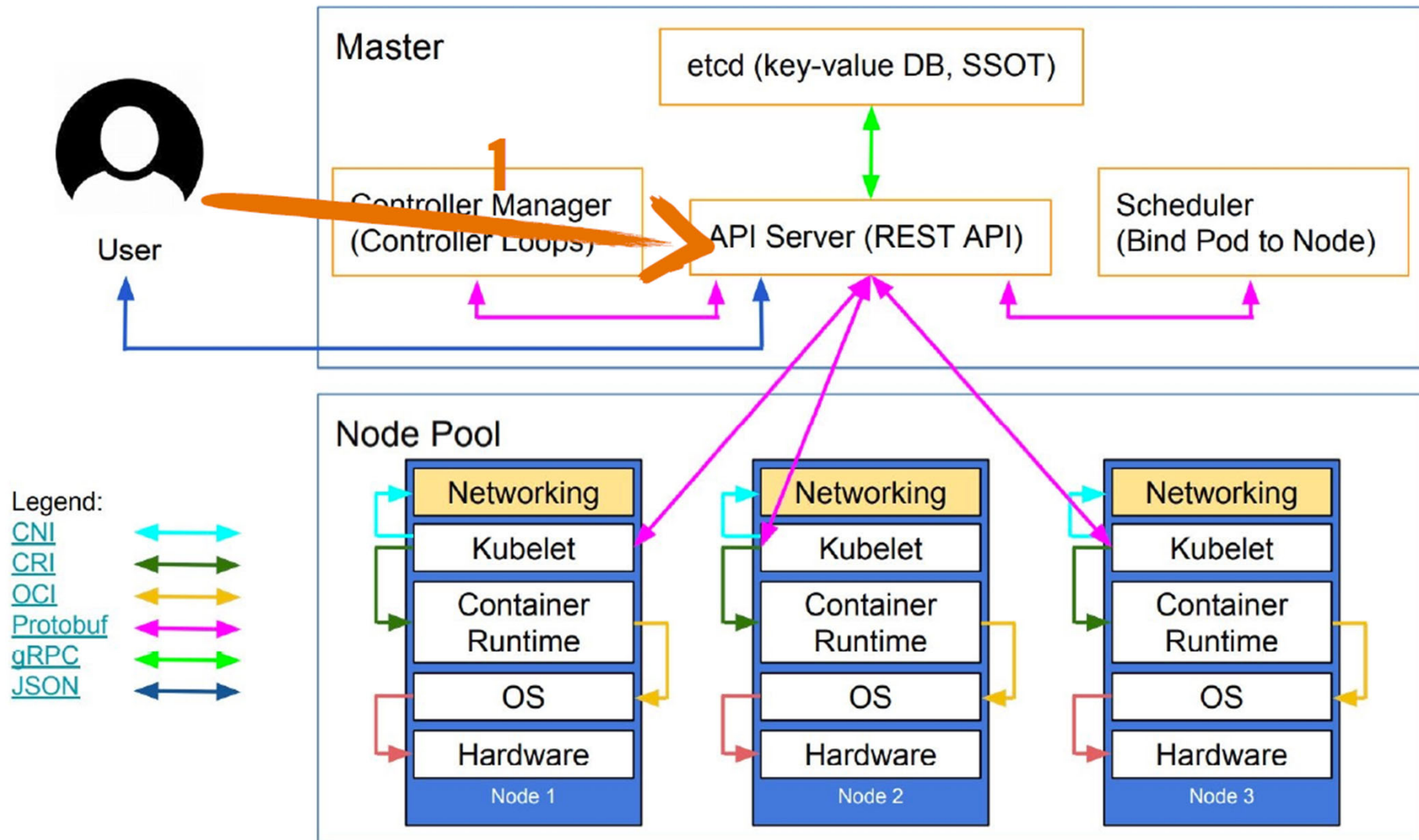


What happen when ...

Imagine I want to deploy a Deployment to a Kubernetes cluster. I'd probably type something like this in my terminal:

```
kubectl apply -f deployment.yaml
```

Kubernetes' high-level component architecture

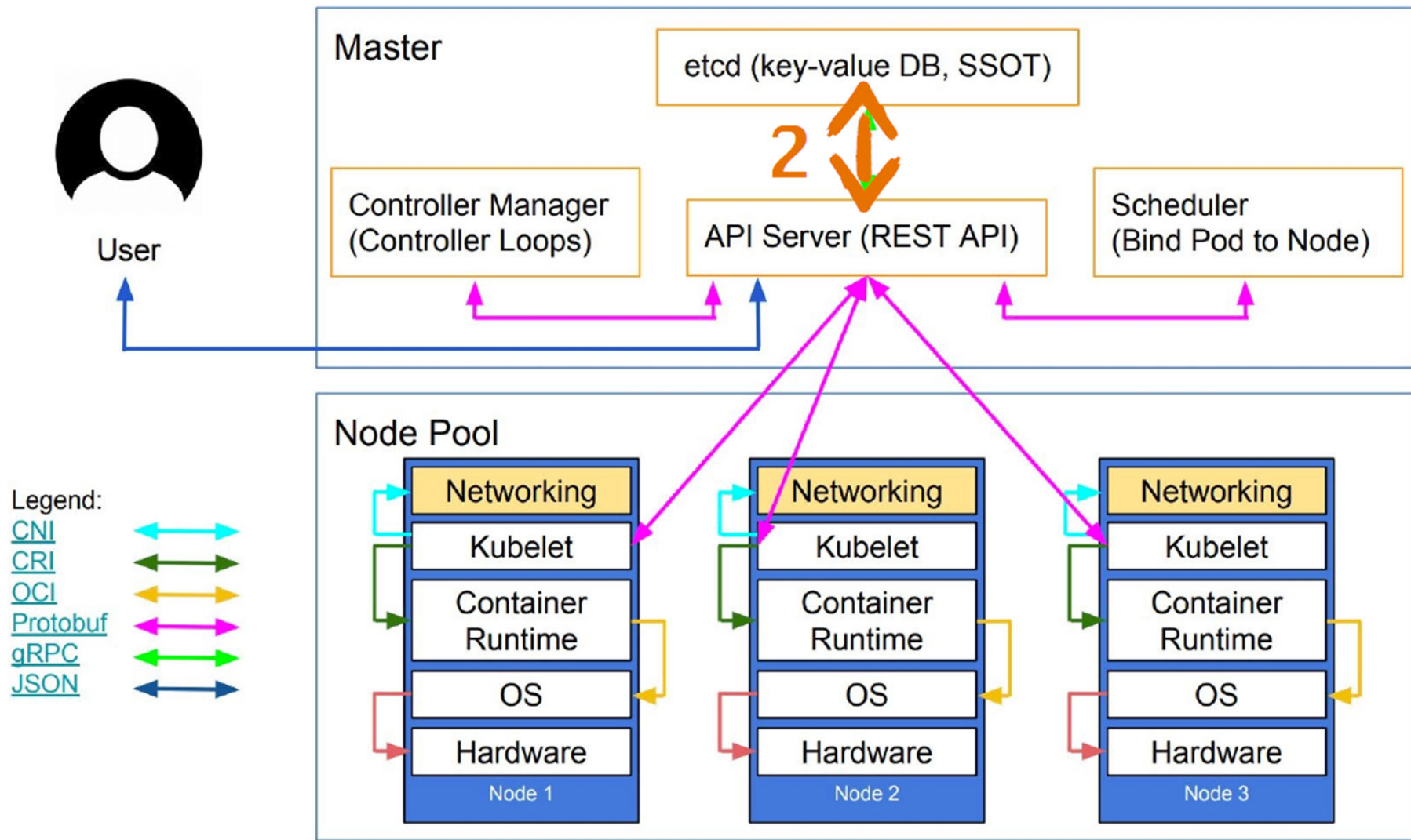


Kubectl

- API Groups: Group, Version, Resource;
- Validate `kubeconfig`;
- Generate and send Request.

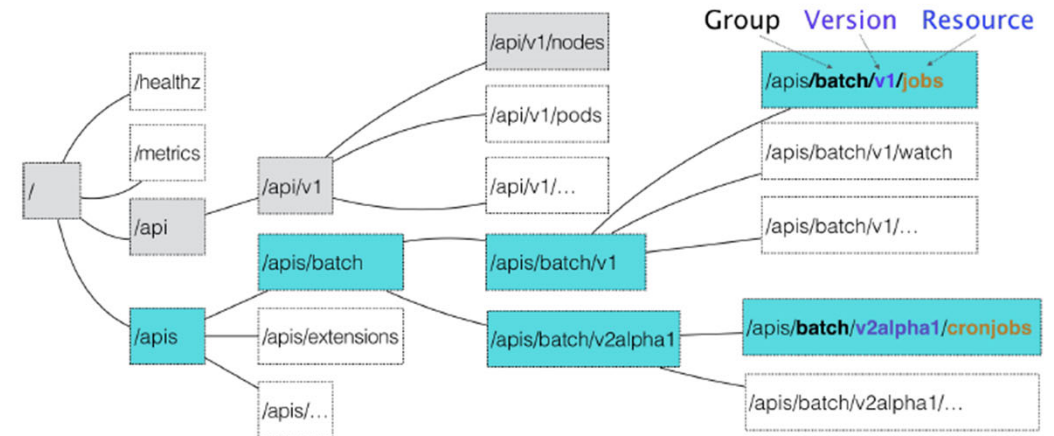
```
+ deployment.yaml
30 apiVersion: apps/v1
29 kind: Deployment
28 metadata:
27   name: minecraft-server
26   namespace: minecraft
25   labels:
24     app: minecraft
23 spec:
22   replicas: 1
21   selector:
20     matchLabels:
19       app: minecraft
18   template:
17     metadata:
16       labels:
15         app: minecraft
14     spec:
13       containers:
12         - name: minecraft
11           image: itzg/minecraft-server
10           resources:
9             requests:
8               cpu: 50m
7               memory: 100Mi
6             limits:
5               cpu: 2
4               memory: 4Gi
3           imagePullPolicy: Always
2           ports:
1             - name: game-port
31               containerPort: 25565
~
```

Kubernetes' high-level component architecture

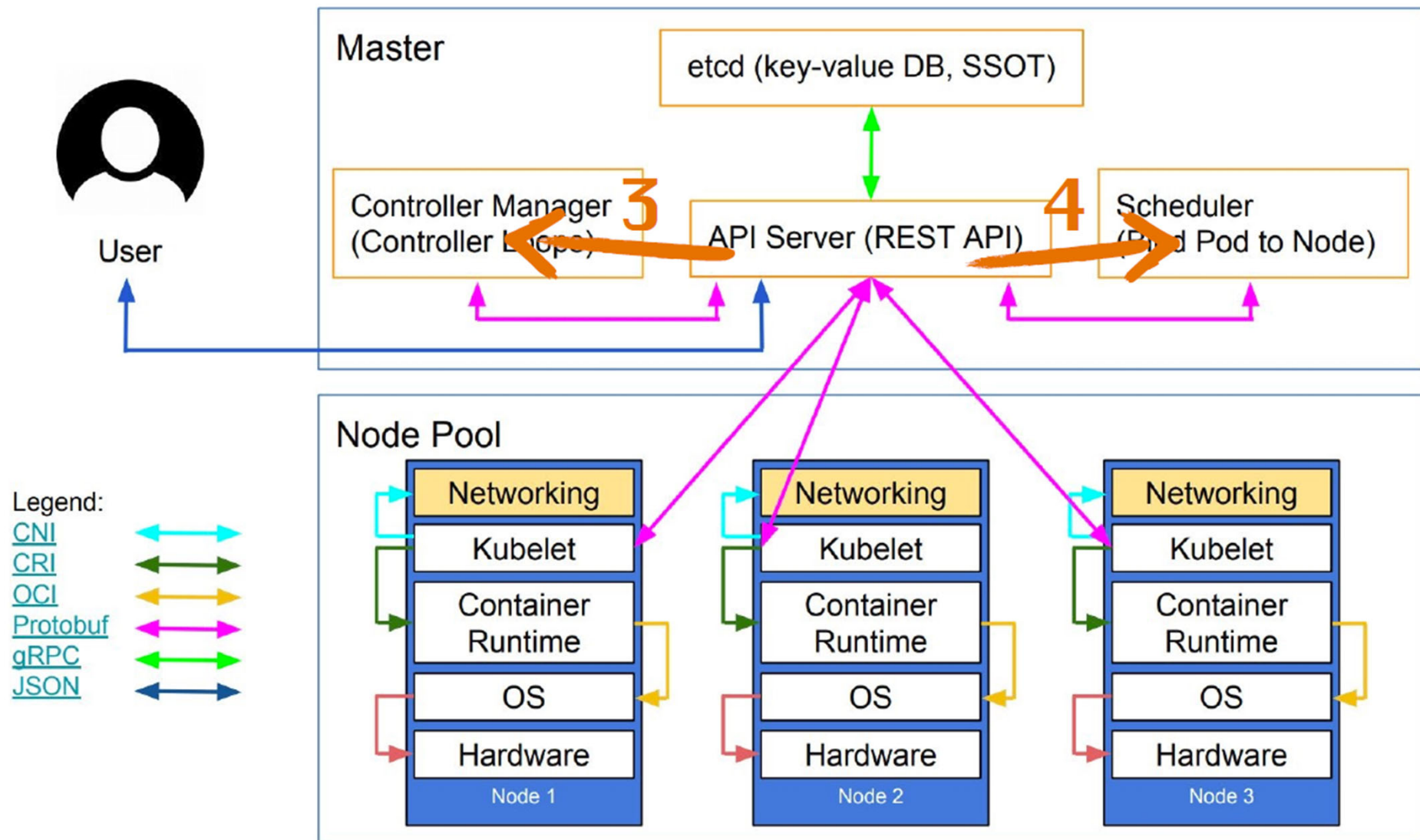


Apiserver

- Authentication & Authorization
- Convert;
- Admission & Validation;
- Call Etcd Api;
- Key format ``<namespace>/<name>``

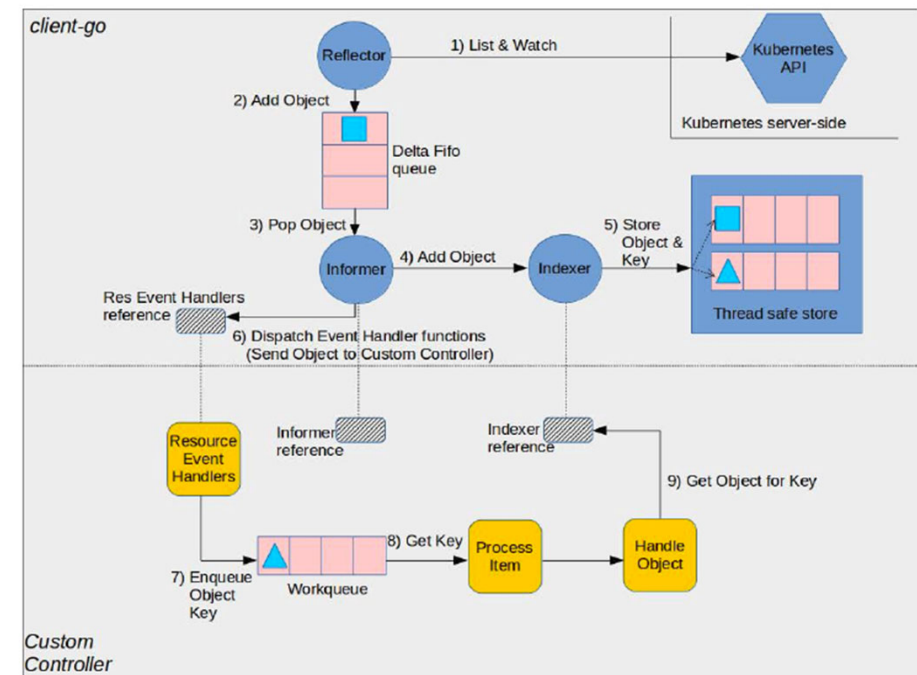


Kubernetes' high-level component architecture



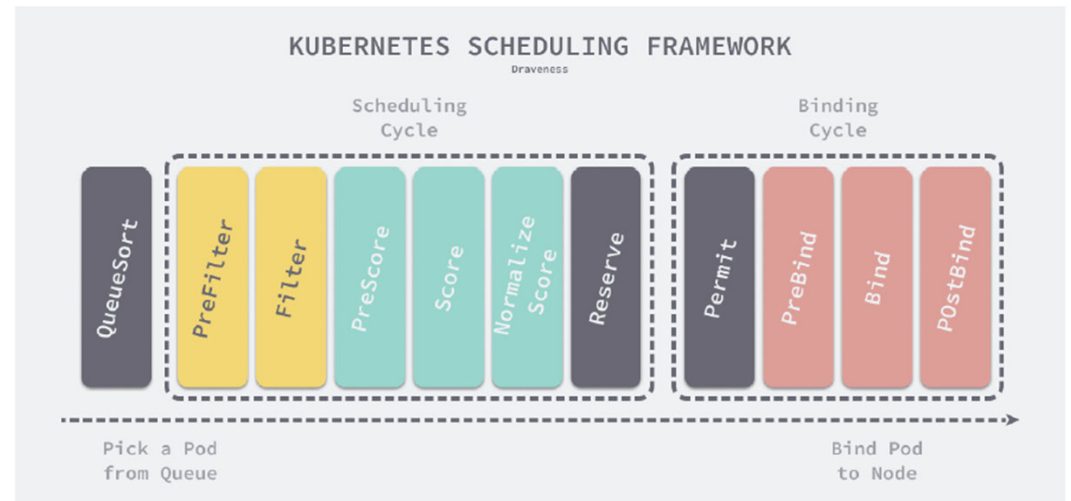
Control loops: Controller Manager

- Controller: current -> desired
 - Deployment Controller
 - Replicaset Controller
- Informer:
 - Reflector;
 - DeltaFIFO;
 - Indexer;
 - Workqueue;

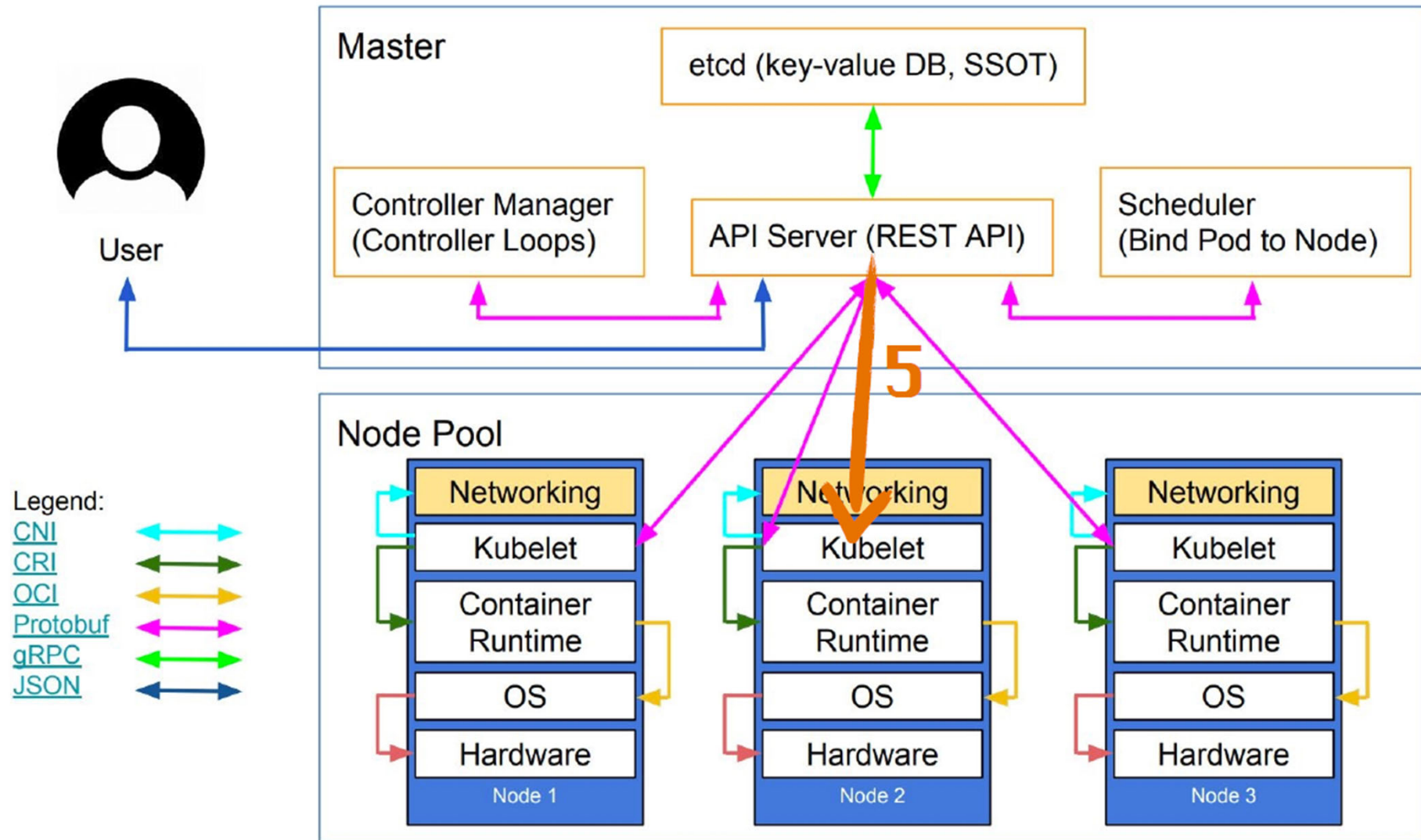


Control loops: Scheduler

- Listen and reconcile;
- Scheduling Cycle;
- Binding Cycle;
 - Send to apiserver;
 - Update etcd.

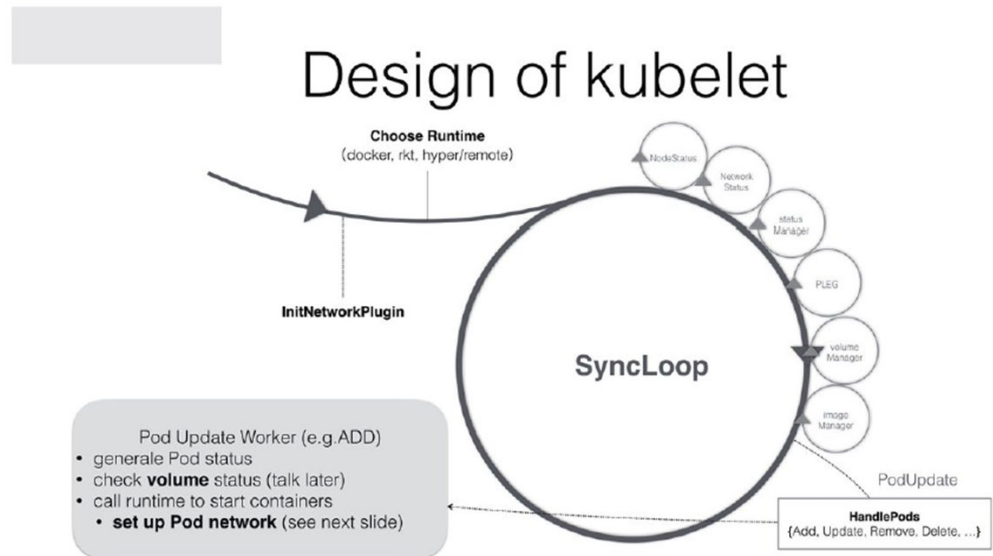


Kubernetes' high-level component architecture



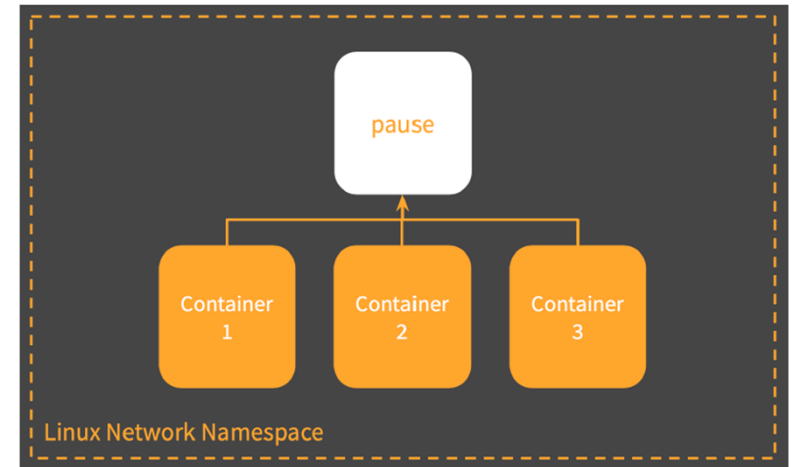
Kubelet: Pod Sync

- Kubelet is a Special Controller;
- Create a Pod:
 - Generates a PodStatus;
 - Cgroup;
 - Create data directories;
 - Attach Volumn;
 - Run Container Runtime



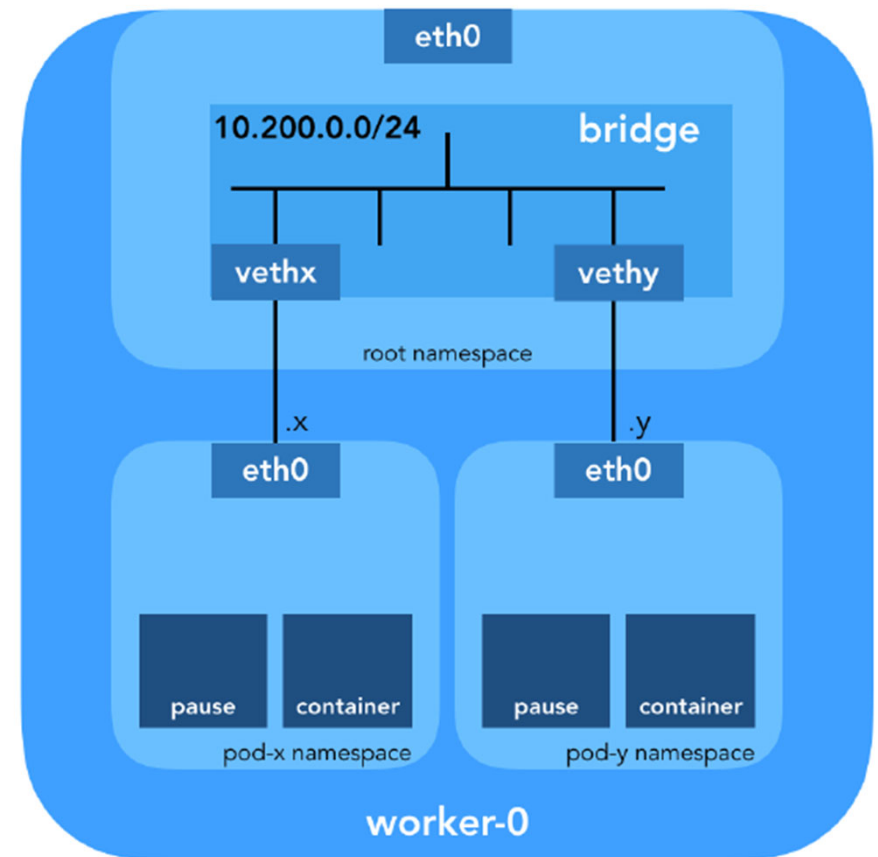
Kubelet: CRI

- CRI Abstraction;
- Pause Container:
 - Share namespace;
 - Init Process.



Kubelet: CRI

- CNI Abstraction;
- Linux Bridge;
- Veth Pair;
- Assign an IP to pause container;
- `resolv.conf` for DNS.



Container Startup

- Pull the image;
- Create the container;
- Start the container.

