

containerd: introduction

with Phil Estes and Derek McGowan

Agenda

- **What is containerd?**
- **Why use containerd?**
- **How to contribute?**
- **What is next?**

(What is containerd?)

- **A “Container runtime”**

- Below platforms (Docker, Kubernetes)
- Above lower level runtimes (runc, Kata, Firecracker, gVisor)

- **Resource Manager**

- Container processes
- Image artifacts
- Filesystem snapshots
- Metadata and dependencies

- **Tightly scoped**

- 100% maintainer approval required to increase scope
- Built-in CRI plugin only scope increase

Highlights

- Feb 2019: 5th project to **graduate** in the CNCF
- Over **250 contributors**; representing 150+ companies
- Supports Linux & Windows; multiple architectures
- Governed sub-projects: Rust ttrpc; encryption, lazy image pull

(History of containerd)

- **Originally built up alongside Docker, not forked or inherited**
- **Grew in scope from a container supervisor to full runtime**
- **Completely new interfaces for managing containers and images**
- **CRI originally out of process plugin brought into core**

Should you use containerd?

YES!

Using Kubernetes? Most stable, Best supported, Most mature
see Most major cloud provider Kubernetes offerings

For Development? Most commonly used tools use containerd to run containers
see Docker, buildkit

For Edge? Most memory efficient and stable
see k3s

For Functions As A Service? Fastest and most efficient
see faasd, IBM Cloud Functions

Stability

1. Process stability

Process does not crash or corrupt data, does not cause containers to crash

2. Runtime resources

Low memory usage, low CPU usage, no large spikes, small container overhead

3. Filesystem usage

Filesystem resources are accounted for and cleaned up by garbage collector

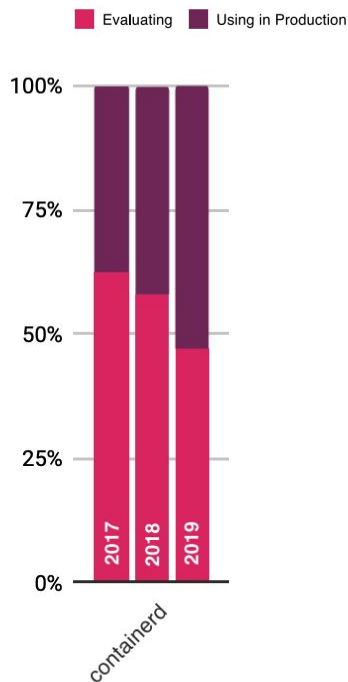
Widely used in production

By:

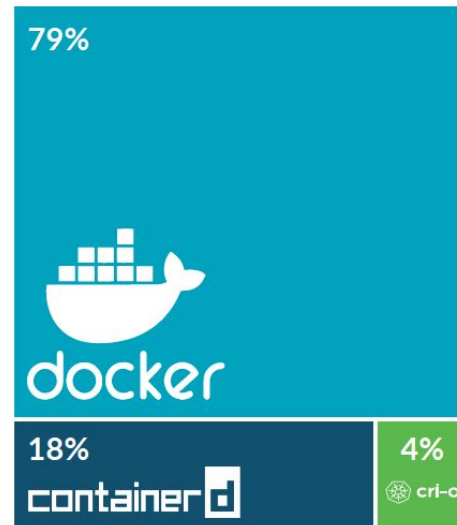
- Cloud Providers
- Kubernetes as a Service
- Docker
- Isolation Technologies
- Containers “aaS”
- Developer Tools (Ignite, Kind, BuildKit, etc.)



Who is using containerd?



- 2018-2019: Now > 50% of containerd users use it in production (**CNCF Annual Survey**, March 2020)
- When adding Docker's use of containerd and pure use of containerd directly, this October 2019 report showed a combined 96% of container usage (**Sysdig 2019 Container Usage Report**, October 2019)



Where are you using it?

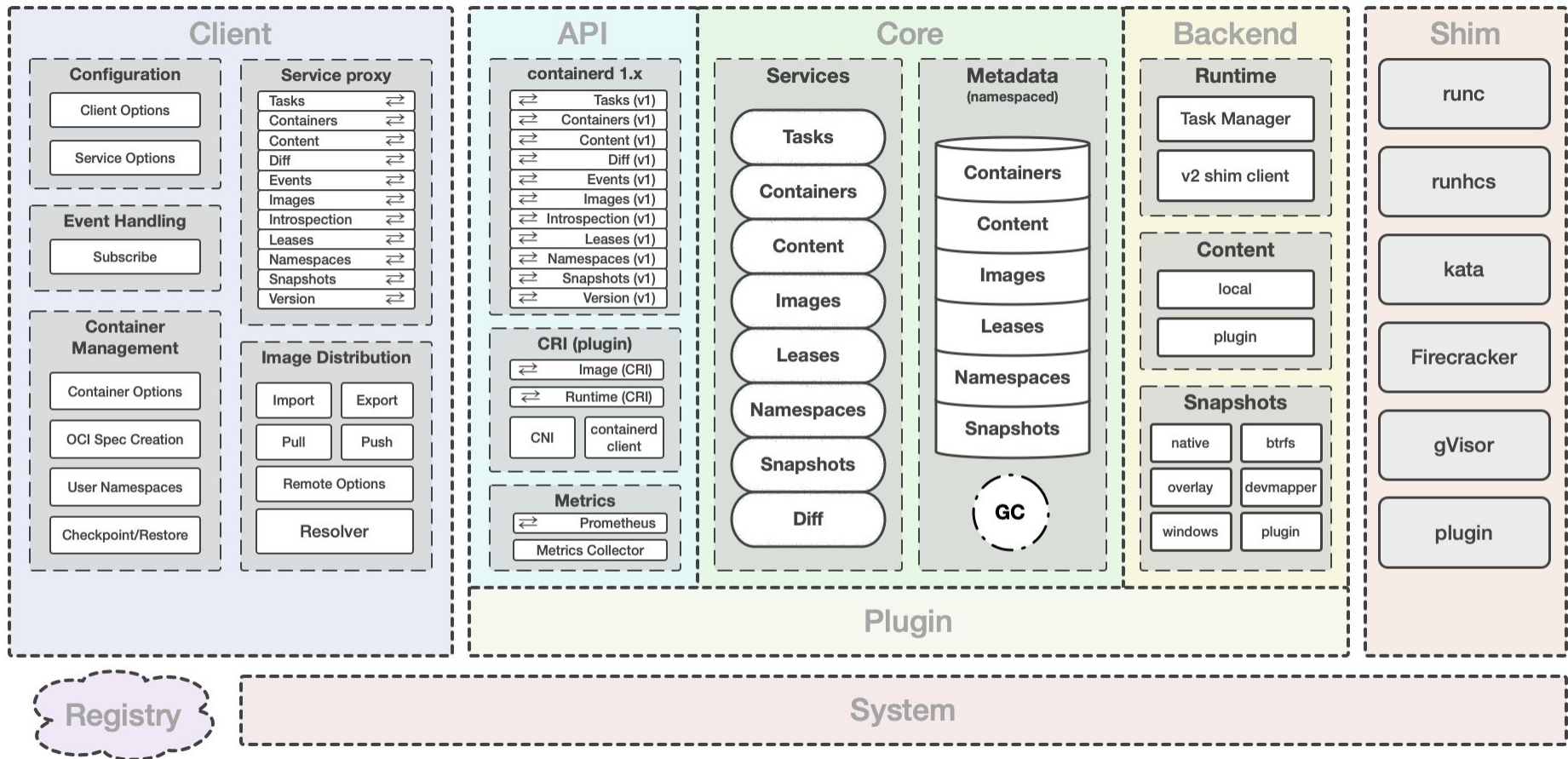
How to contribute

File Issues

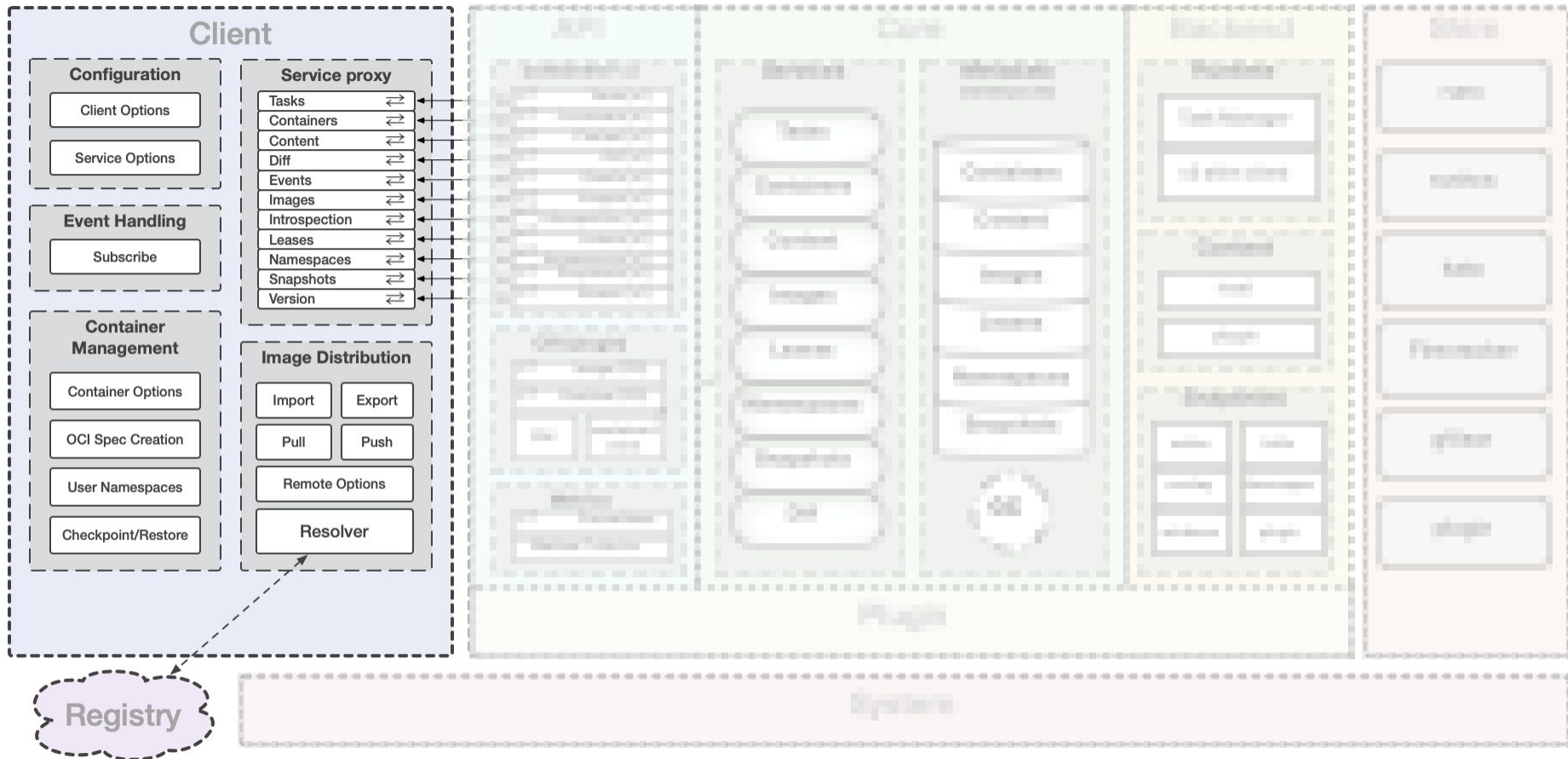
Join Discussions

Reach out on Slack

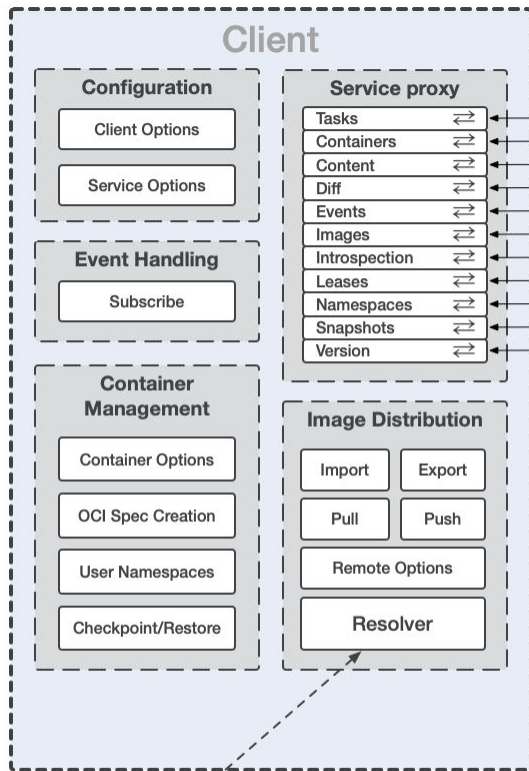
Overview



Client



Client - Options



```
type ServicesOpt
```

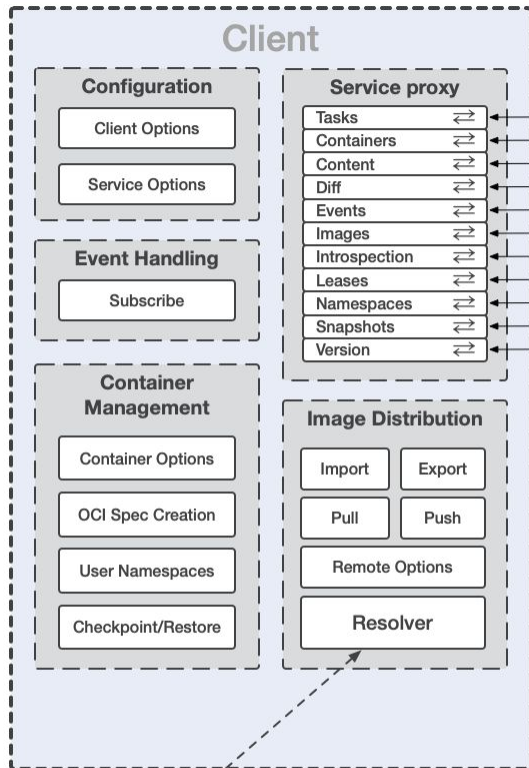
```
func WithContainerService(containerService containersapi.ContainersClient) ServicesOpt
func WithContentStore(contentStore content.Store) ServicesOpt
func WithDiffService(diffService diff.DiffClient) ServicesOpt
func WithEventsService(eventService EventService) ServicesOpt
func WithImageService(imageService imagesapi.ImagesClient) ServicesOpt
func WithLeasesService(leasesService leases.Manager) ServicesOpt
func WithNamespaceService(namespaceService namespacesapi.NamespacesClient) ServicesOpt
func WithSnapshotters(snapshotters map[string]snapshots.Snapshotter) ServicesOpt
func WithTaskService(taskService tasks.TasksClient) ServicesOpt
```

```
type RemoteOpt
```

```
func WithImageHandler(h images.Handler) RemoteOpt
func WithImageHandlerWrapper(w func(images.Handler) images.Handler) RemoteOpt
func WithResolver(resolver remotes.Resolver) RemoteOpt
```

Registry

Client - Labels



```
// ListStatuses returns the status of any active ingestions whose ref match the  
// provided regular expression. If empty, all active ingestions will be  
// returned.
```

```
ListStatuses(ctx context.Context, filters ...string) ([]Status, error)
```

```
// Commit commits the blob (but no roll-back is guaranteed on an error).  
// size and expected can be zero-value when unknown.  
// Commit always closes the writer, even on error.  
// ErrAlreadyExists aborts the writer.
```

```
Commit(ctx context.Context, size int64, expected digest.Digest, opts ...Opt) error
```

func WithLabels

```
func WithLabels(labels map[string]string) Opt
```

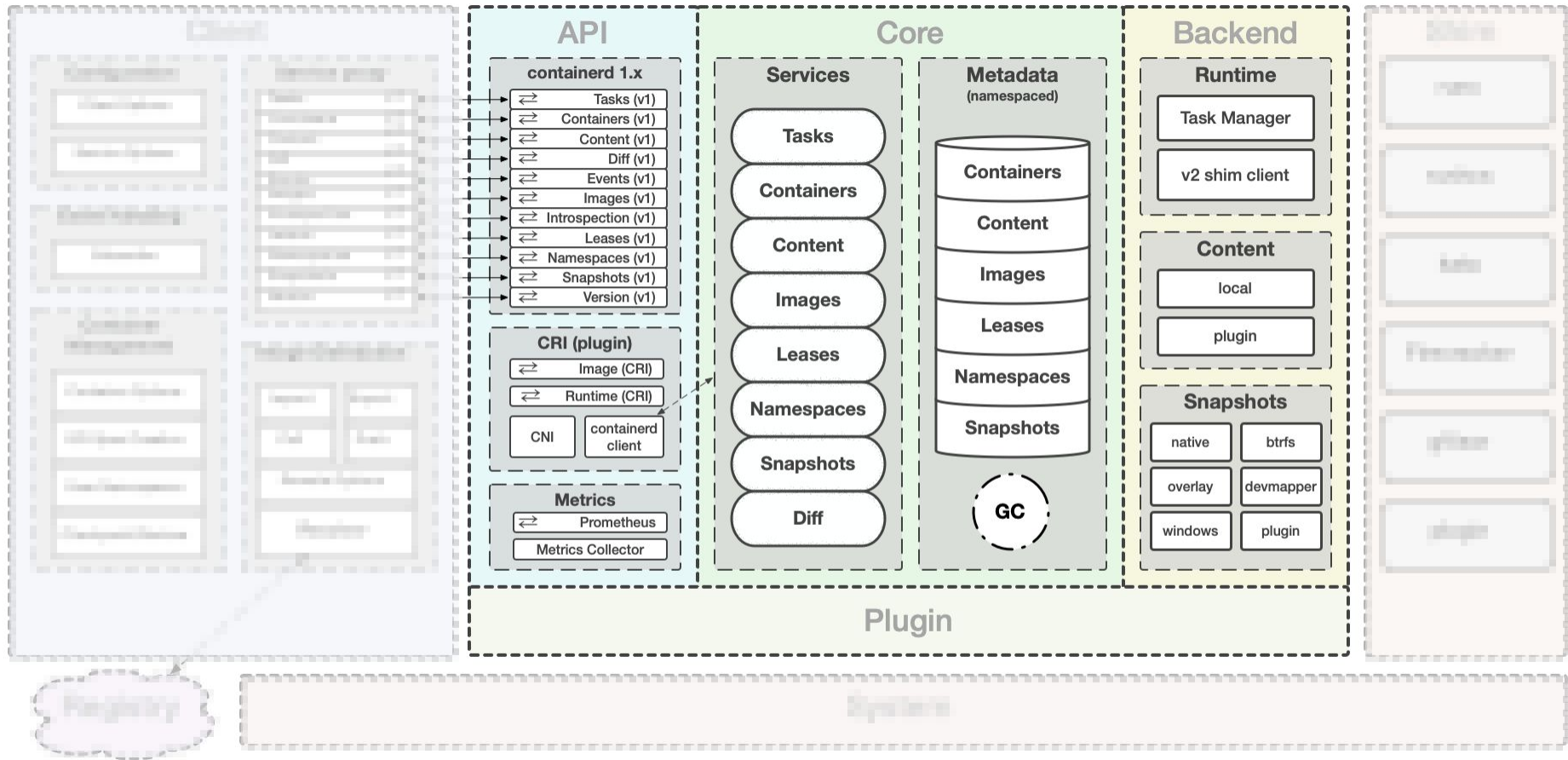
func WithContainerLabels

```
func WithContainerLabels(labels map[string]string) NewContainerOpts
```

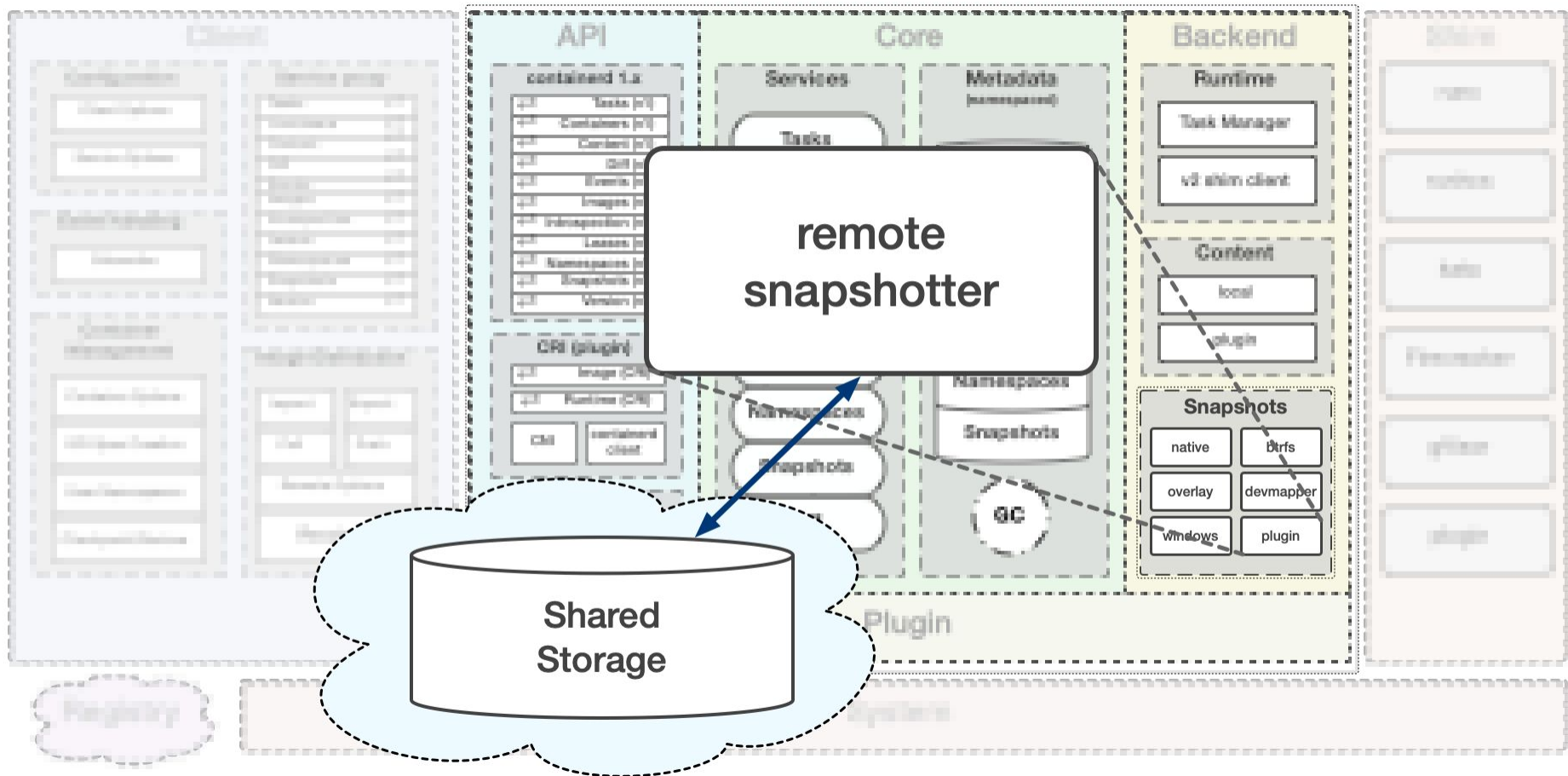
func WithPullLabels

```
func WithPullLabels(labels map[string]string) RemoteOpt
```

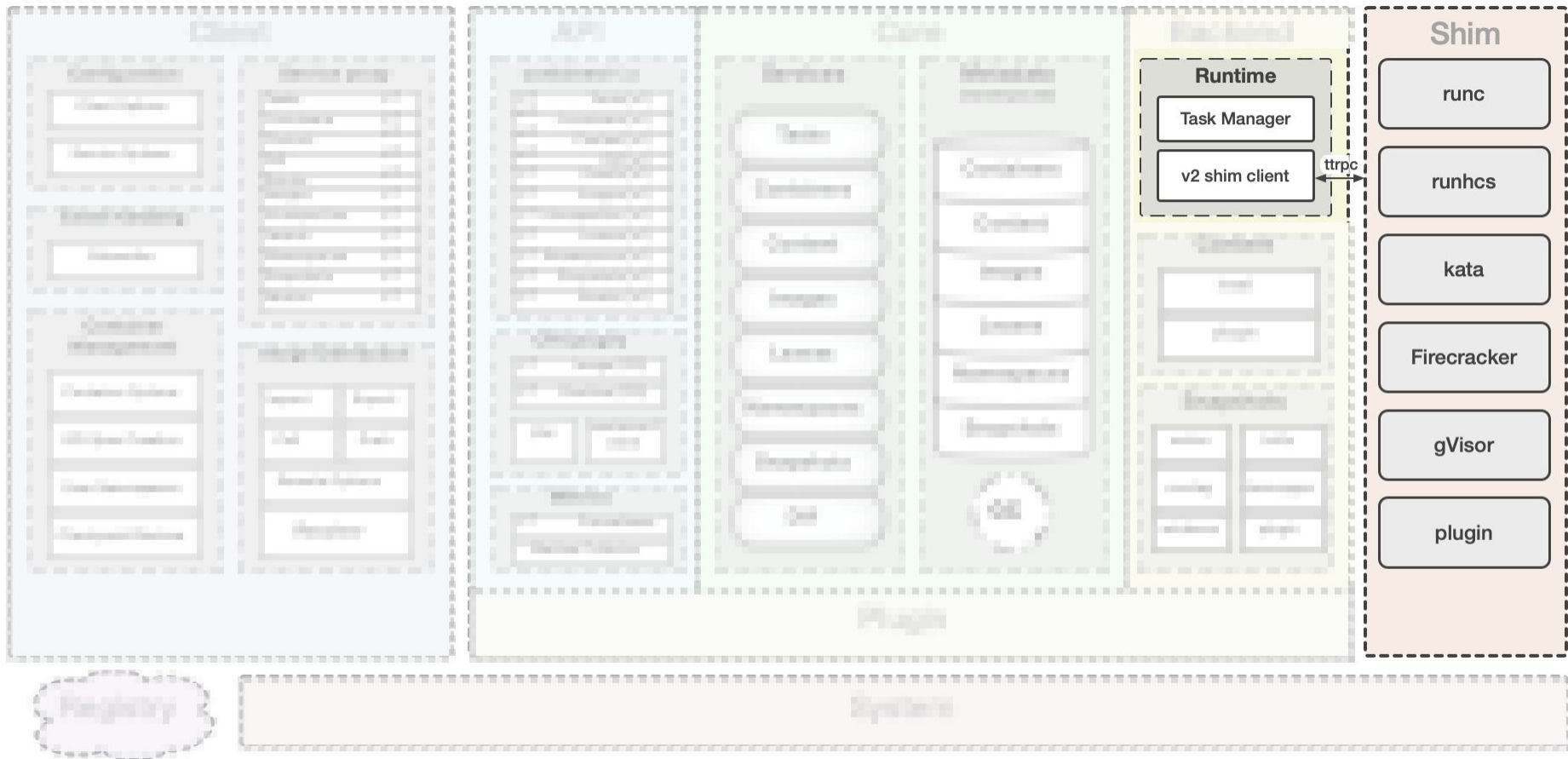

Core



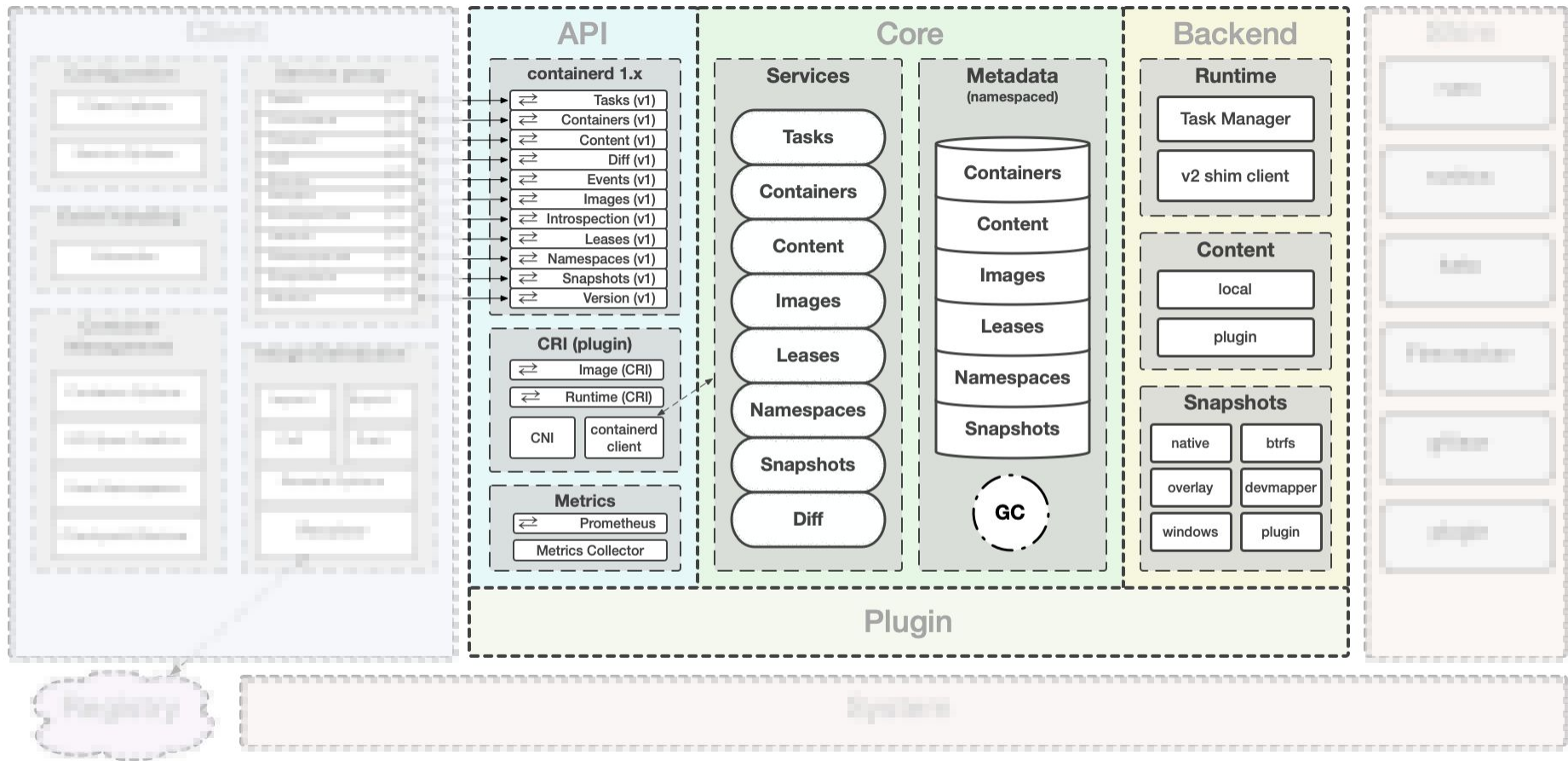
Core



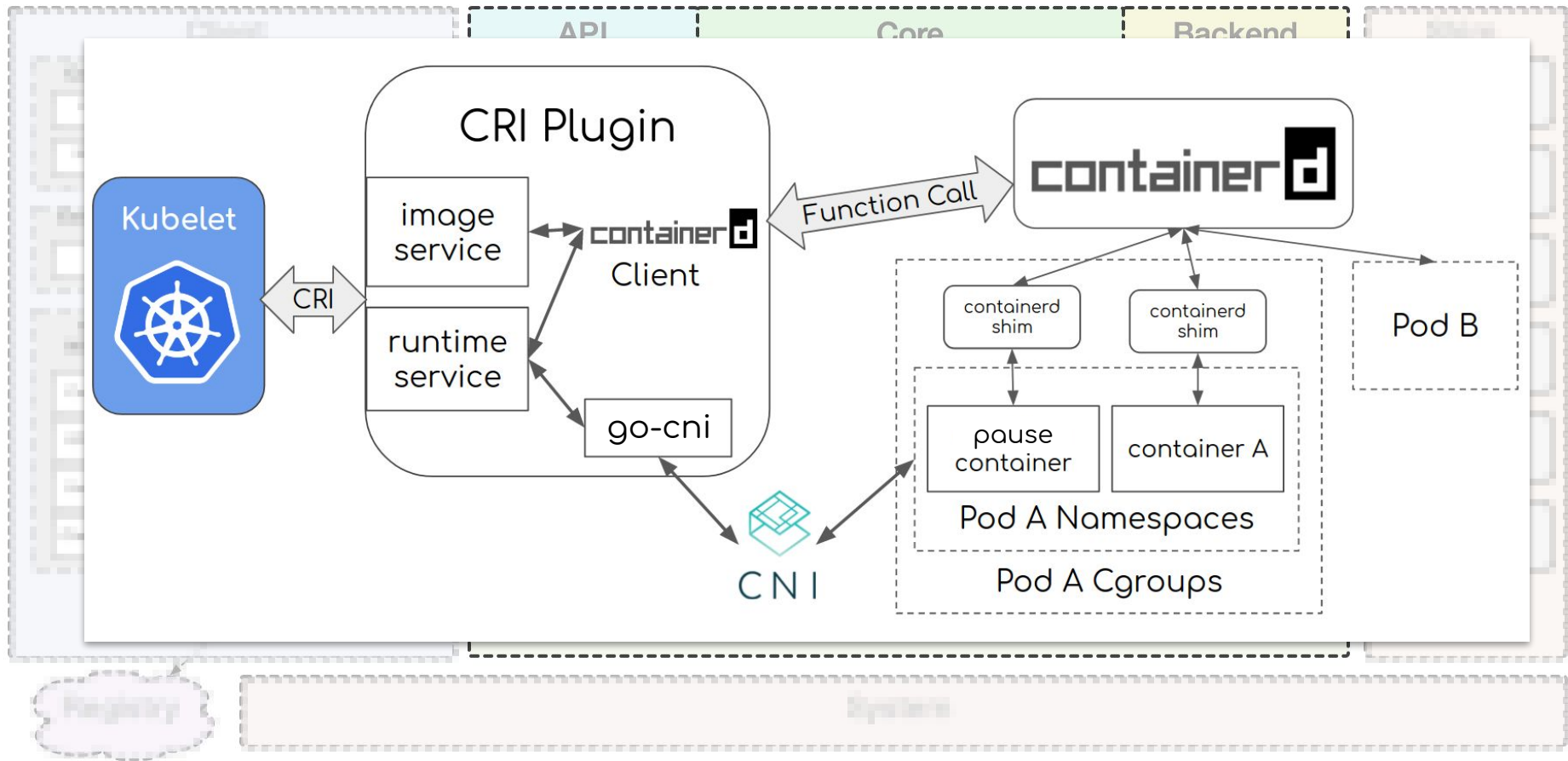
Runtime



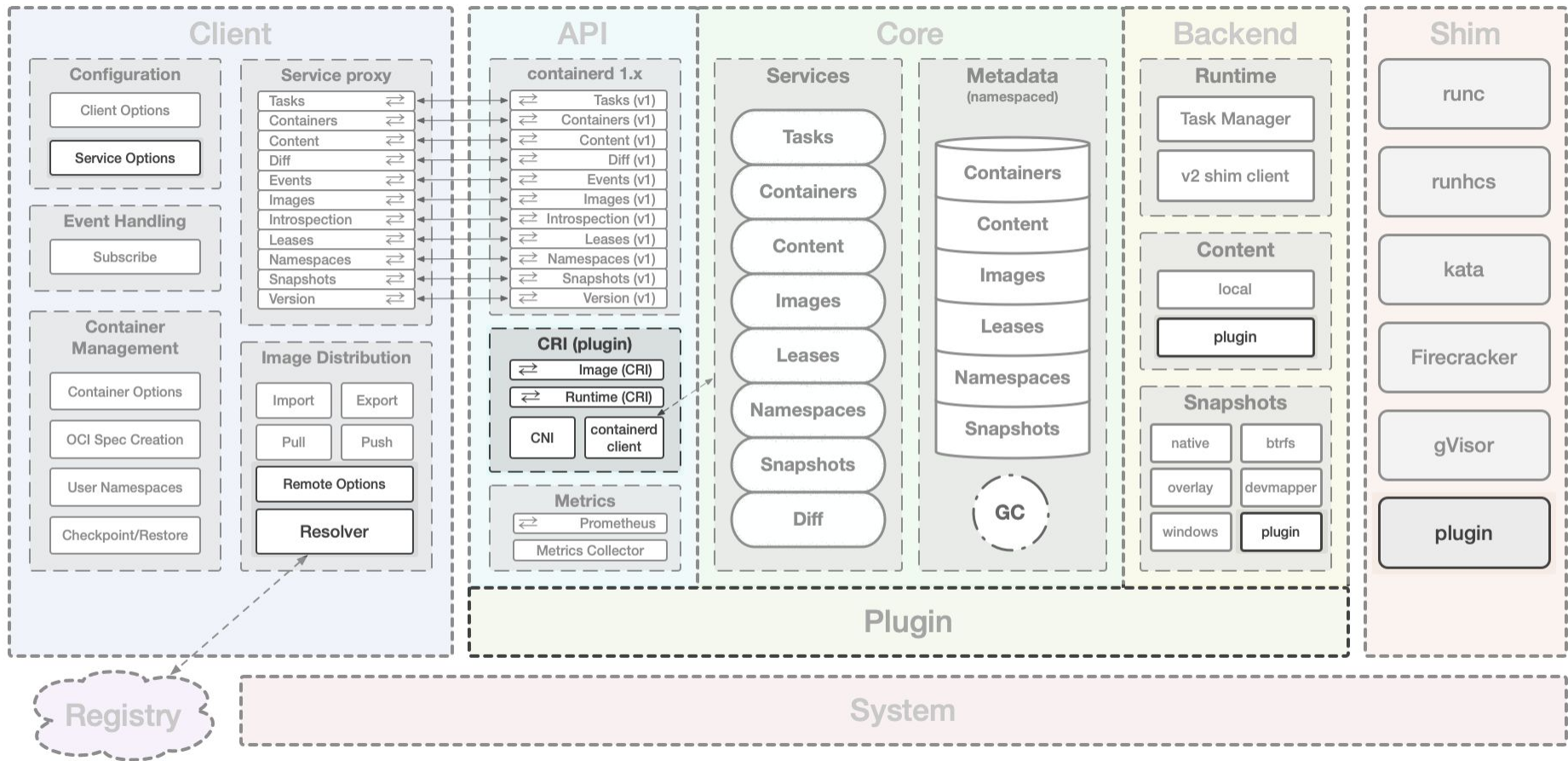
Plugins



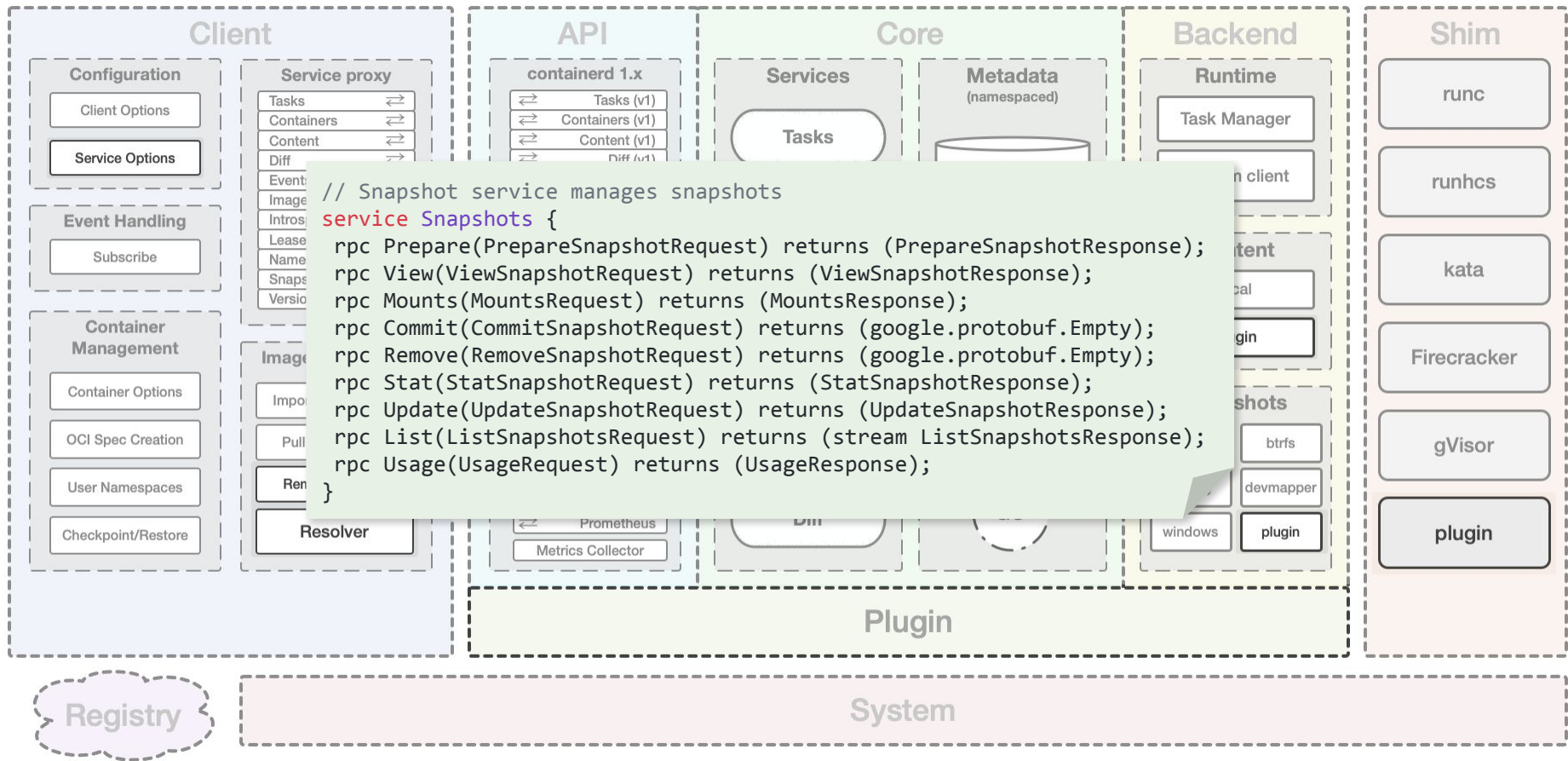
Plugins



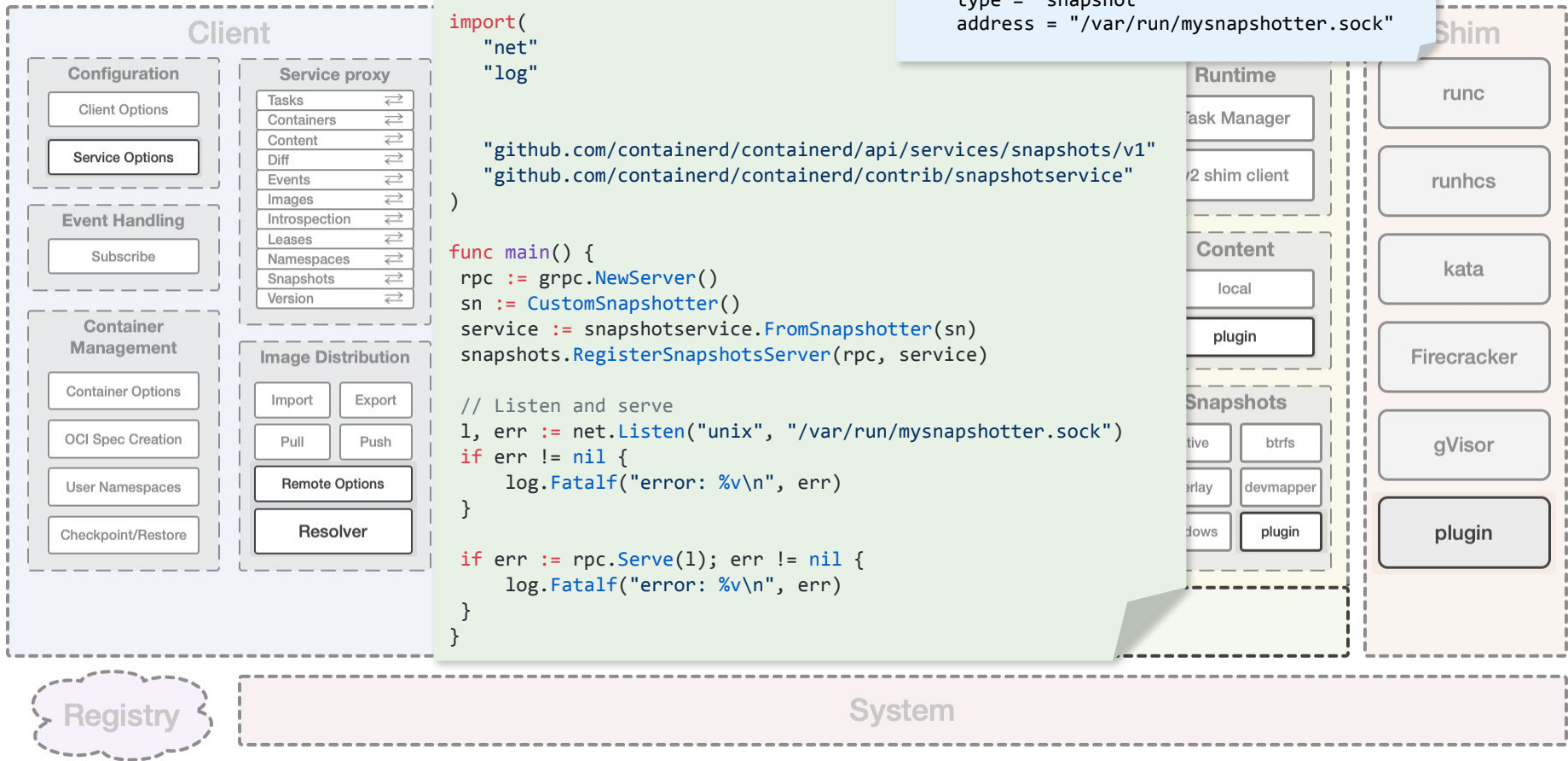
Plugins

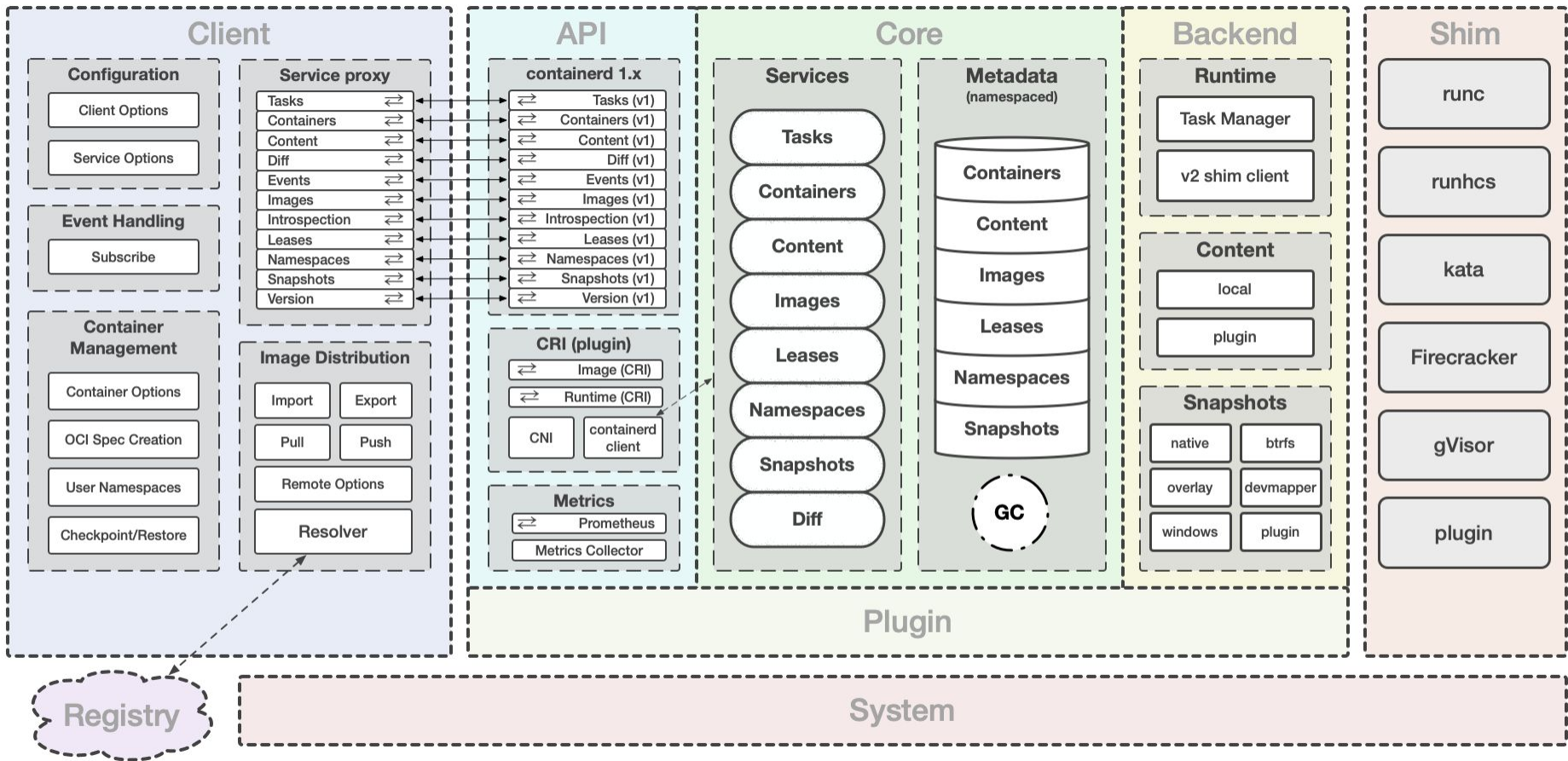


Plugins



Plugins





containerd 1.4 release

- cgroups v2 support
- Improved selinux support
- Support for remote snapshotters
- Windows CRI

containerd 1.5 roadmap

- CRI and containerd code merge
- Sandbox API
- Mount management API

containerd 1.6+

- NRI (Node Resource Interface)
- Improved Kubernetes integration, maybe CRI v2

Thank you!