How Symlinks Pwned K8s

Michelle Au, Software Engineer, Google
Jan Šafránek, Software Engineer, Red Hat

Agenda

Discovery

Development

Disclosure

Secure Practices

Future

Vulnerability Reporting

Reporting

Github issue created 2017-11-30

PodSecurityPolicy can be sidestepped with innocent emptyDir and subpath

Here is a pod which would be allowed by fairly strict security policies, yet gives full control over node host by gaining access to docker socket:

…

Reporting

Github issue created 2017-11-30

PodSecurityPolicy … sidest… with innocent emptyDir and subp…

Here is a pod which … ict security policies, yet gives full control over n… st by gainin… ess to docker socket:

…

Reporting

That's not how it's done!

Follow https://kubernetes.io/docs/reference/issues-security/security/

- Responsibly disclose to allow time to fix before public disclosure
- security@kubernetes.io (optionally GPG encrypted)
- Product Security Team handles the rest
  - Evaluate impact
  - Request CVE
  - Coordinate development of fix, release, disclosure

Vulnerability Details

Volumes Background

Node

```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```

Volumes Background

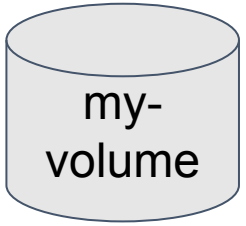

```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```

Node

Volumes Background

Node

```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```

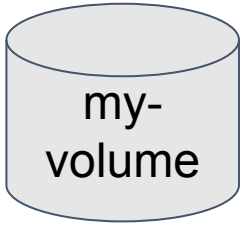


Pod volume's root path (host):
/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume

Volumes Background

Node

```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```



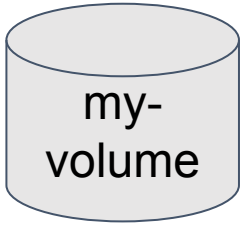
Final host's path:

<pod volume's root path> + <subpath>

Volumes Background

Node

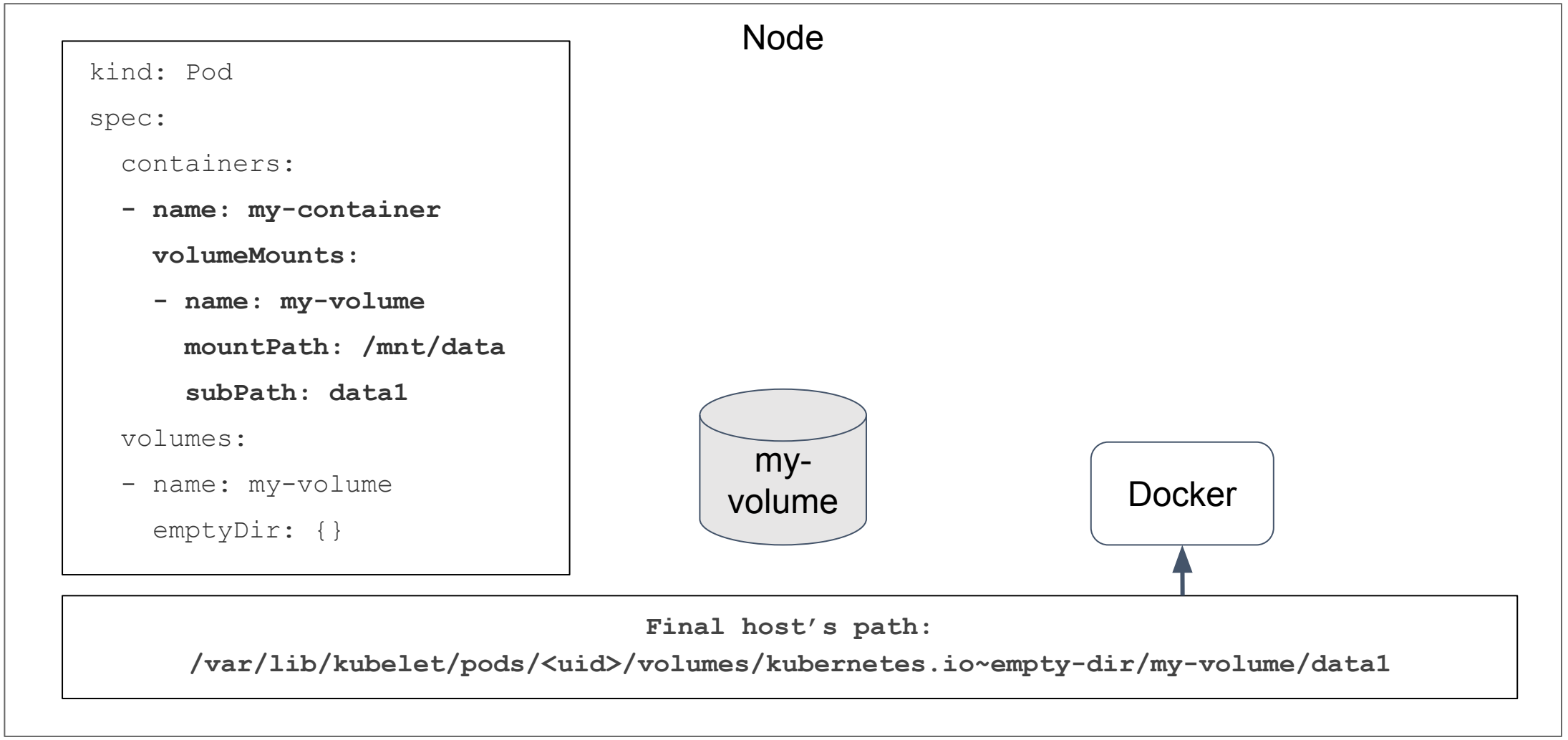
```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```



Final host's path:

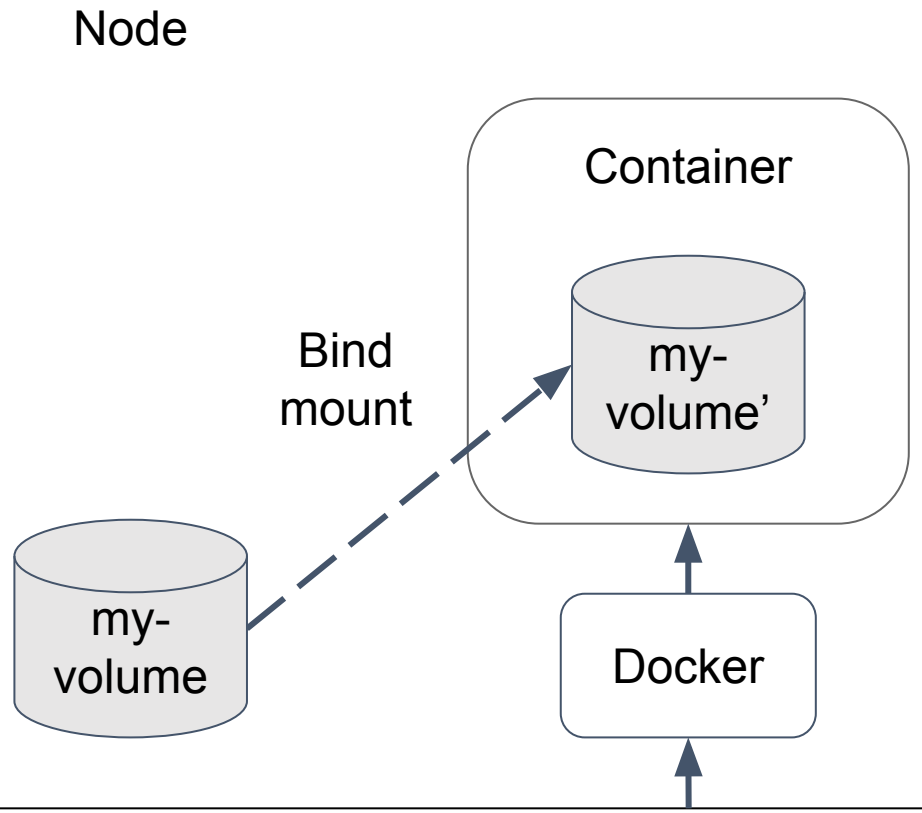
/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Volumes Background



Volumes Background


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```

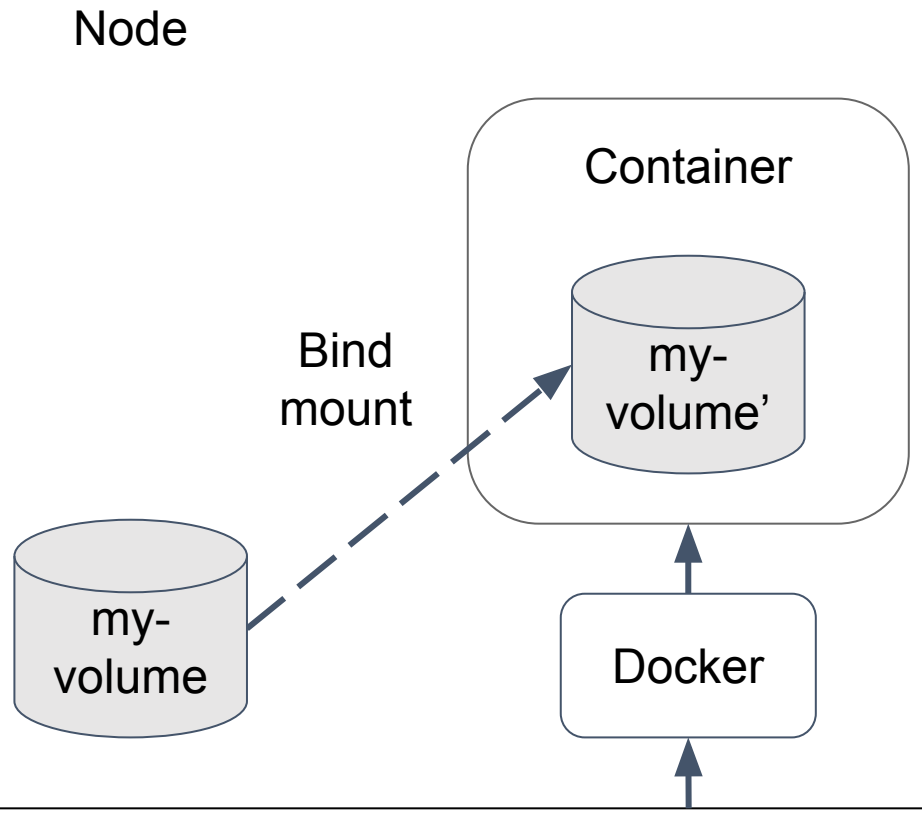


Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Vulnerability?


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```

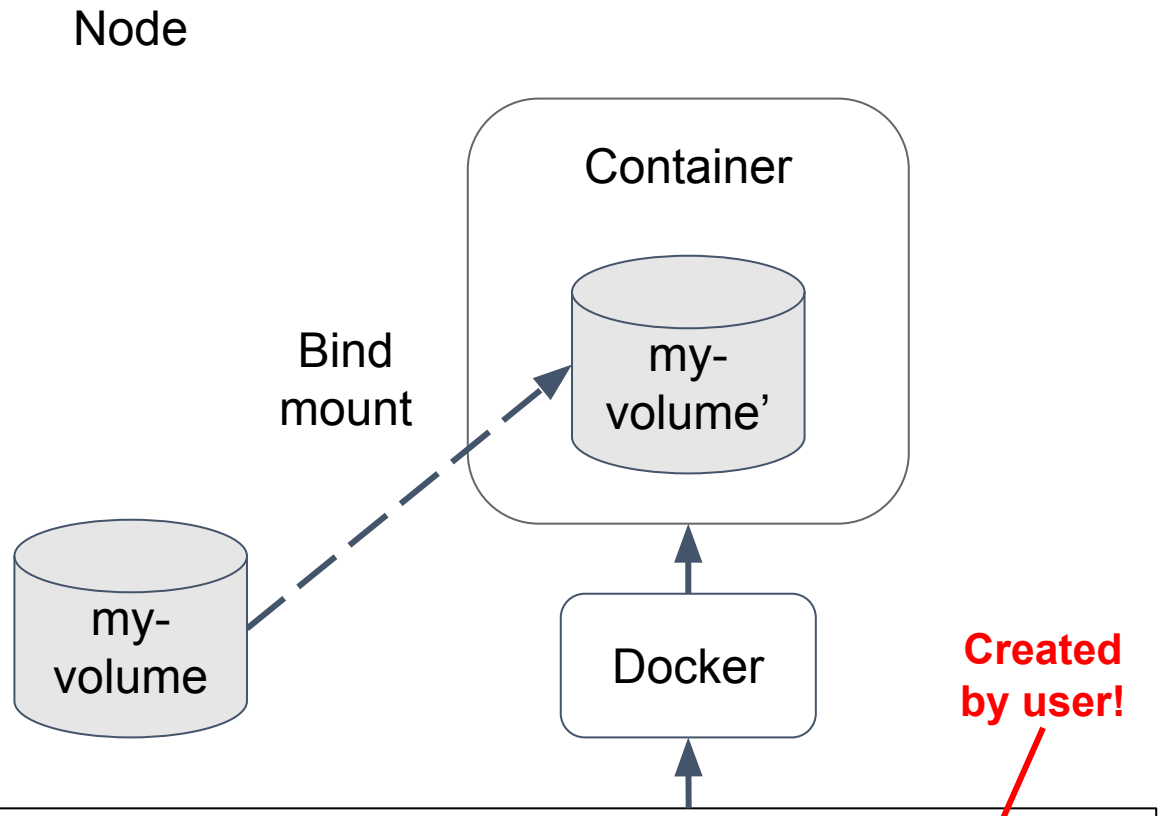


Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Vulnerability


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```



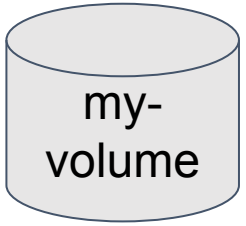
Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Vulnerability

Node

```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```



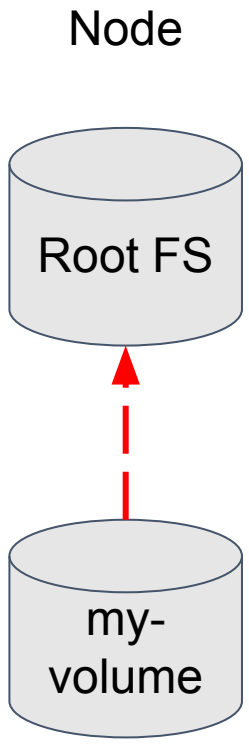
Symlink

Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Vulnerability


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```



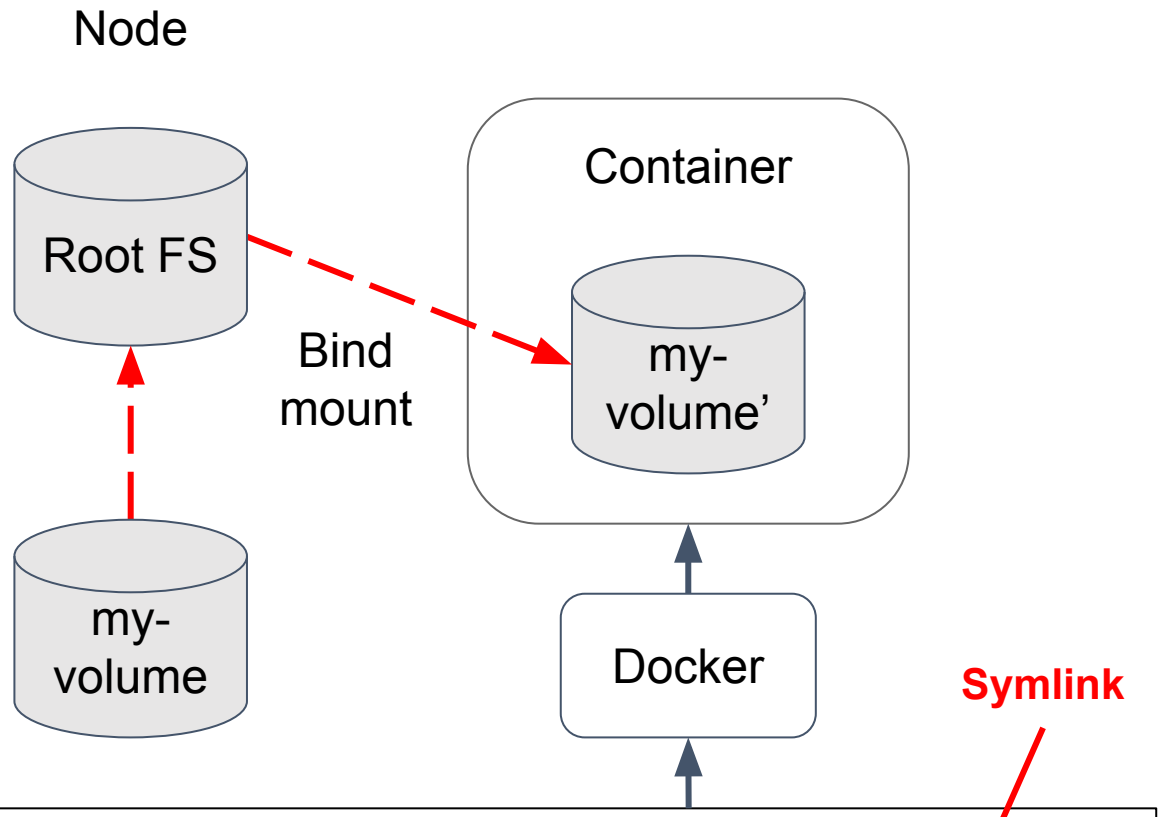
Symlink

Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Vulnerability


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```



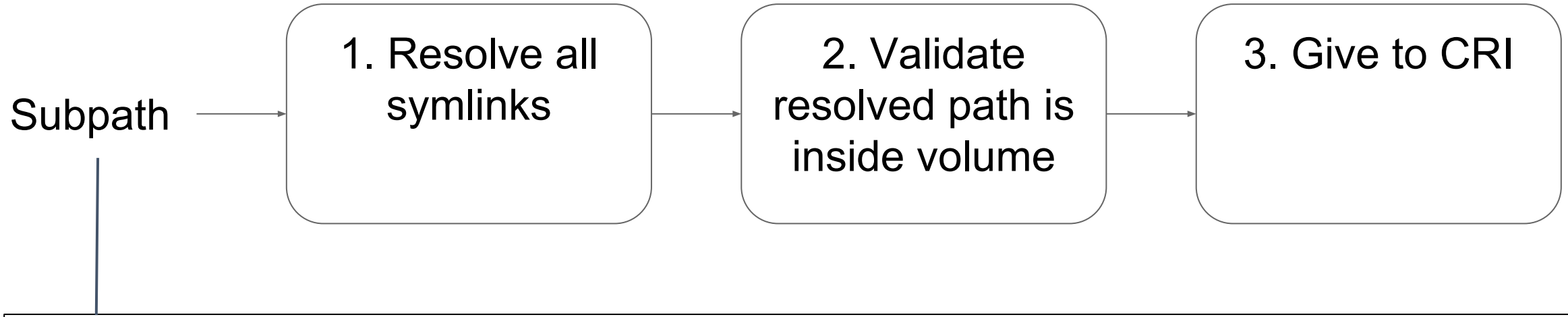
Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

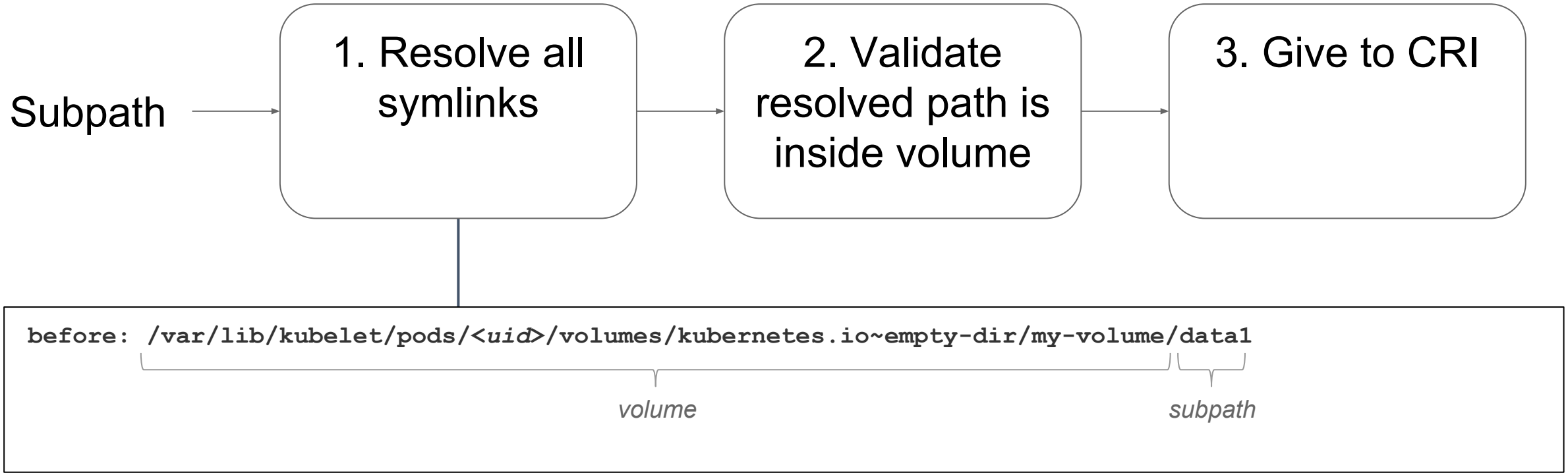
Demo

Solutions

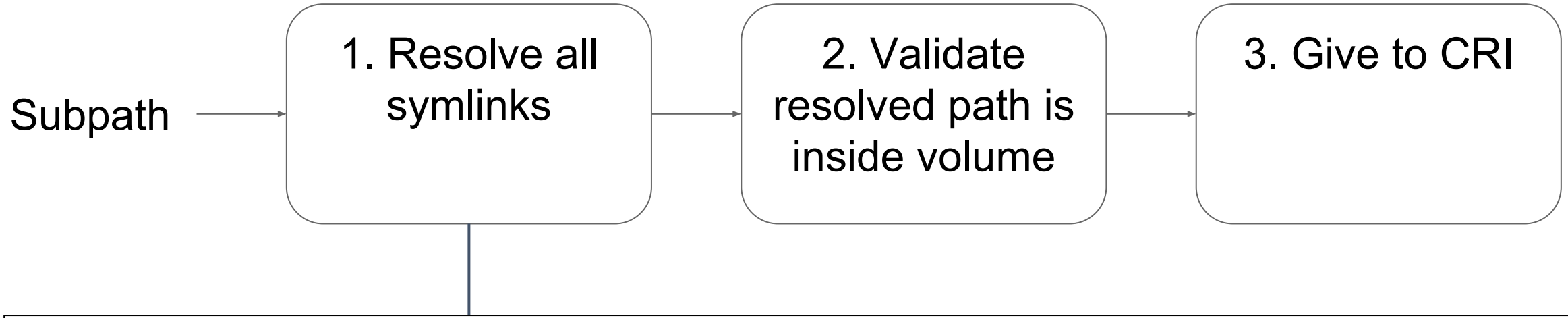
Naive Solution



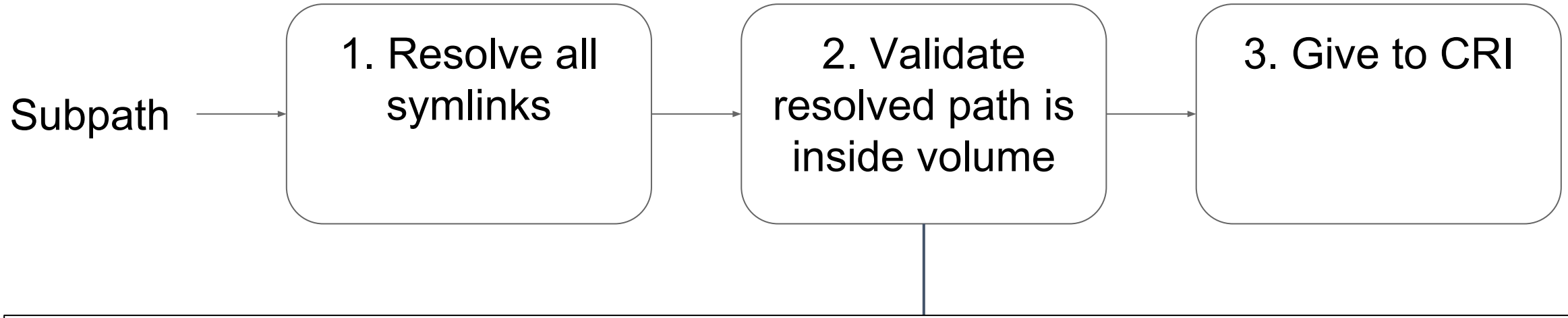
Naive Solution



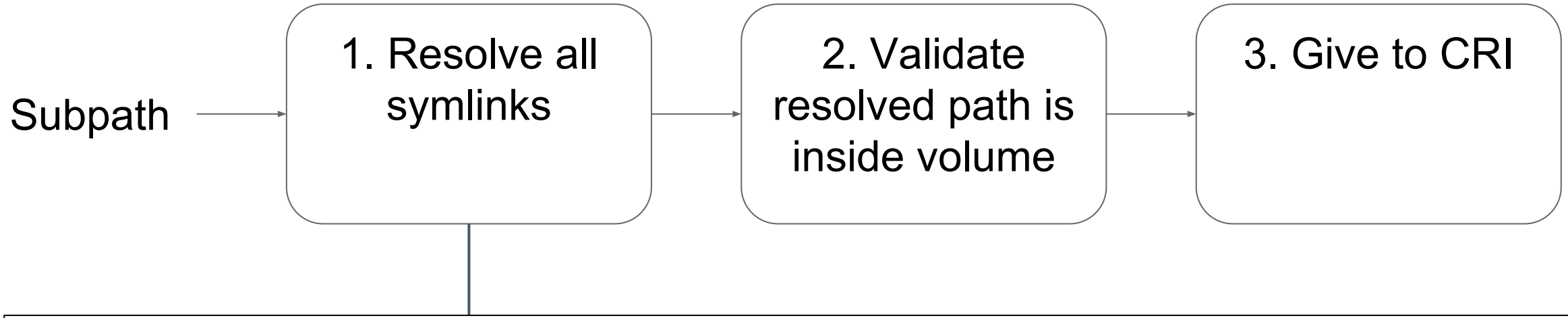
Naive Solution



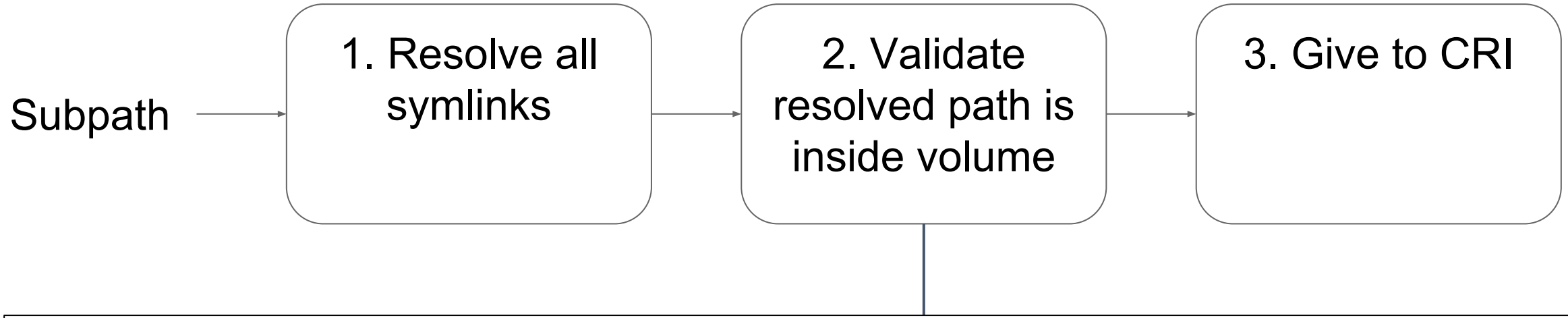
Naive Solution



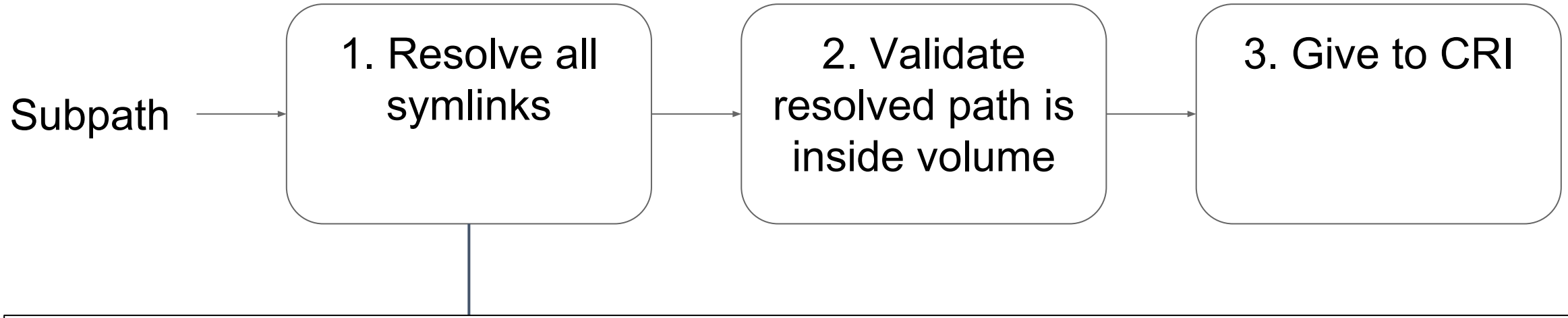
Naive Solution



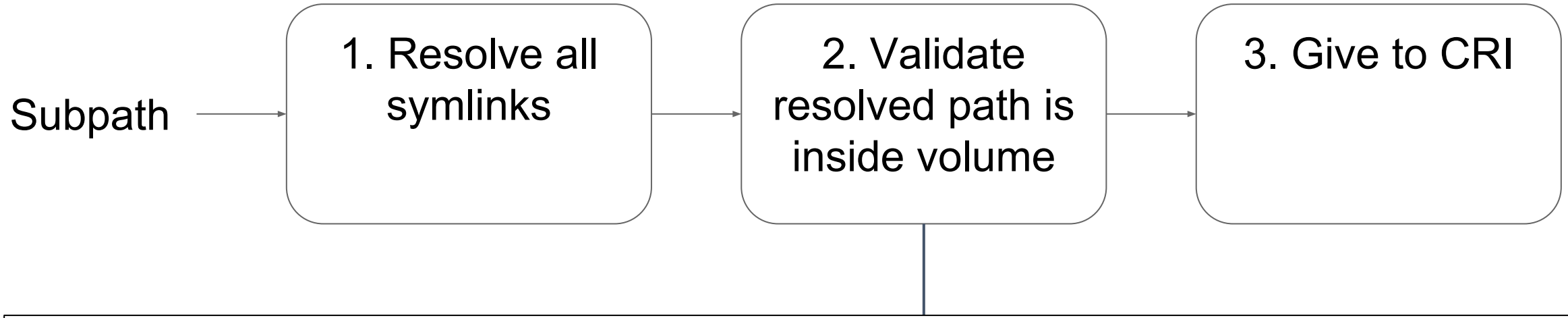
Naive Solution



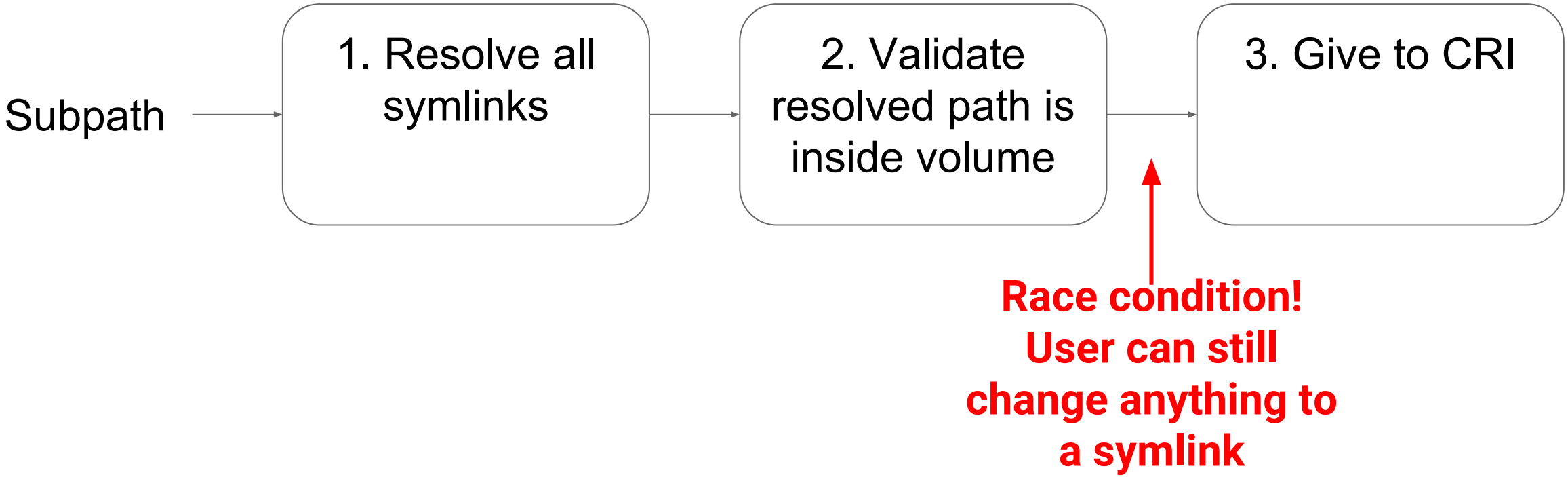
Naive Solution



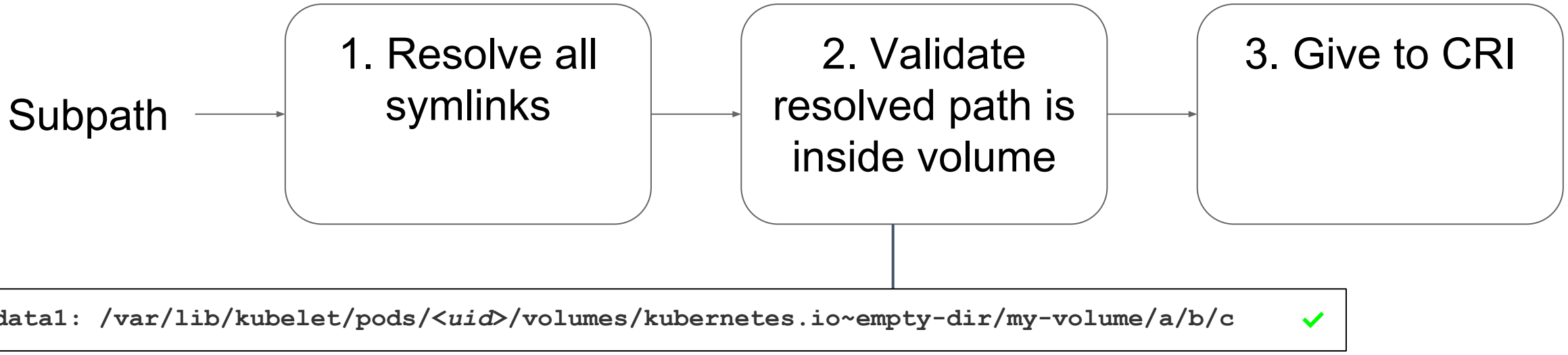
Naive Solution



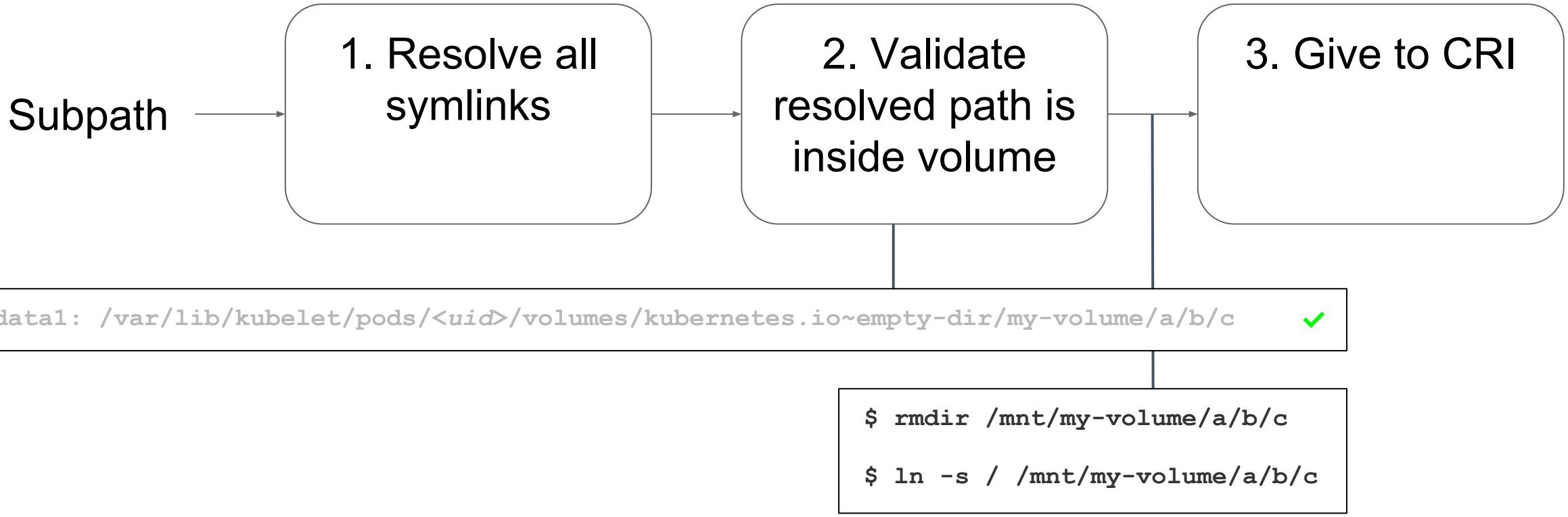
Naive Solution



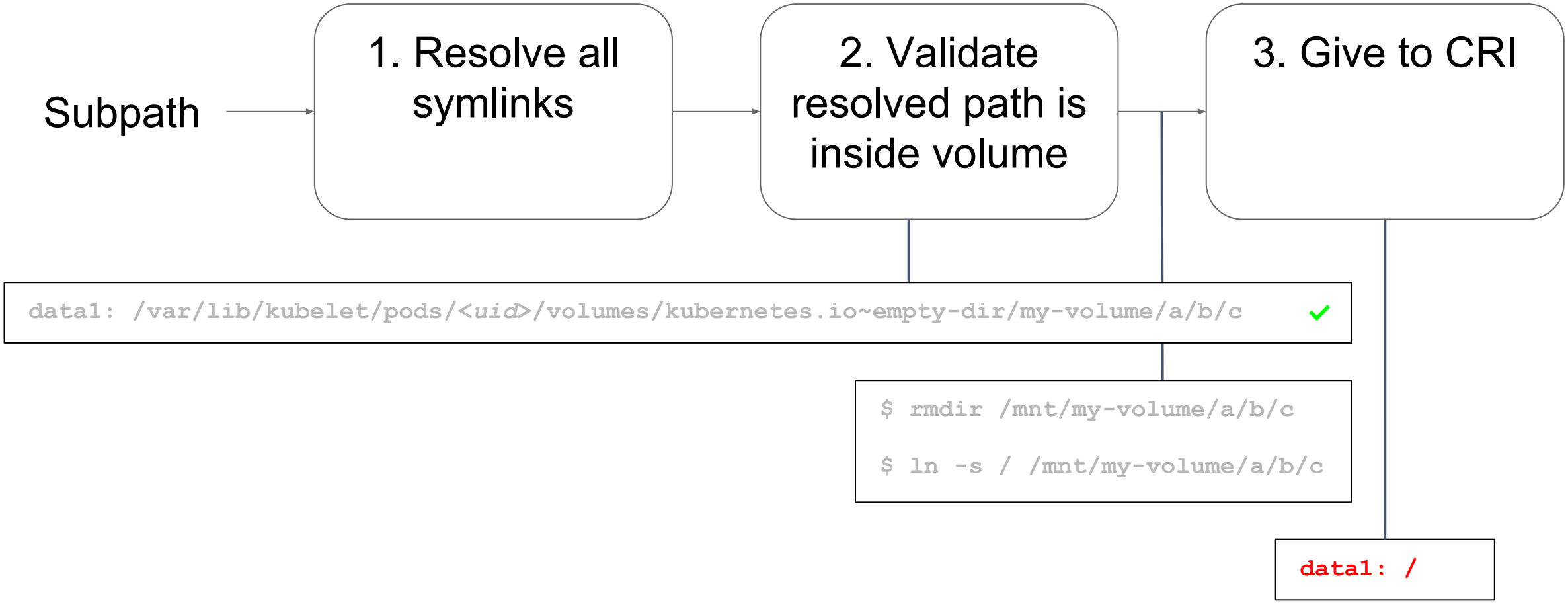
Naive Solution



Naive Solution



Naive Solution



What Now?

Need to “lock” directory between validation and CRI

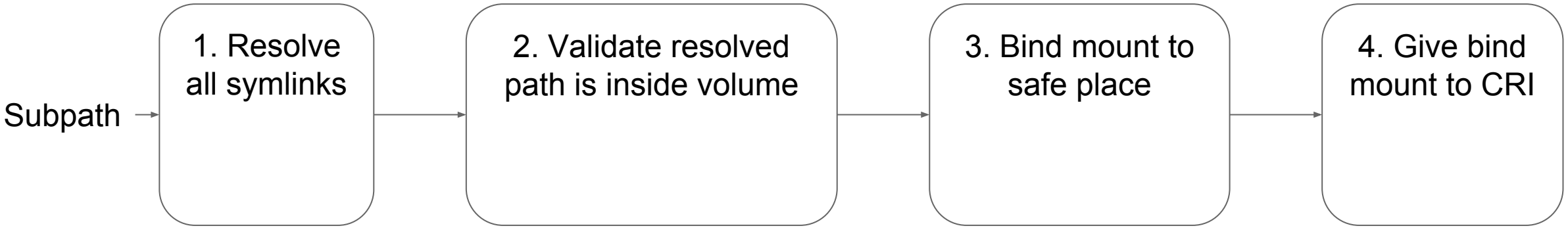
- Windows: lock using `CreateFile`
- Linux: bind mount

Bind mount

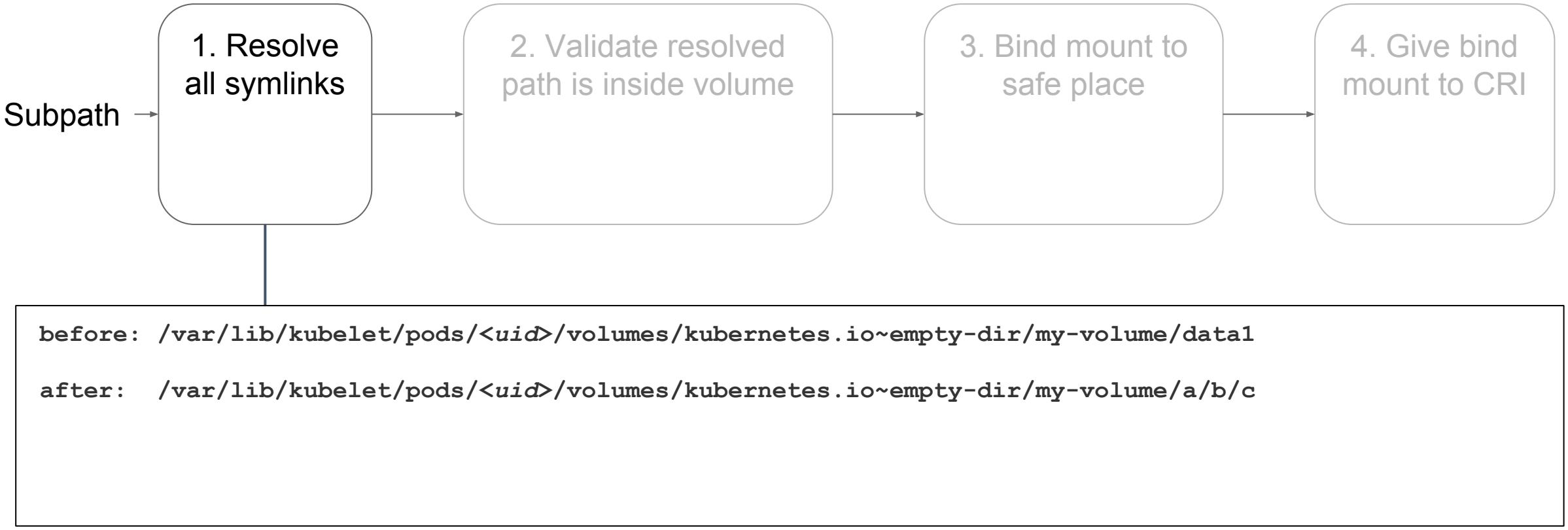
- Remount part of the file hierarchy somewhere else
- Independent on the original hierarchy
- Atomic

```
$ mount --bind \
    /var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/a/b/c \
    /var/lib/kubelet/safe/place
```

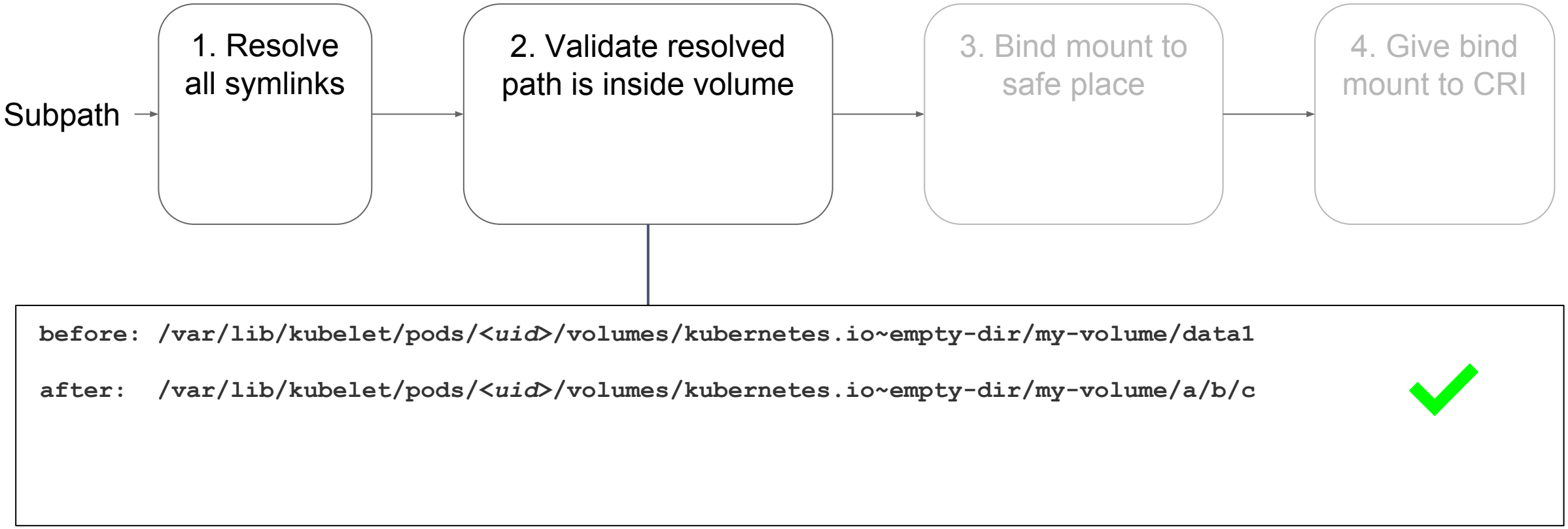
Naive Bind-mount Solution



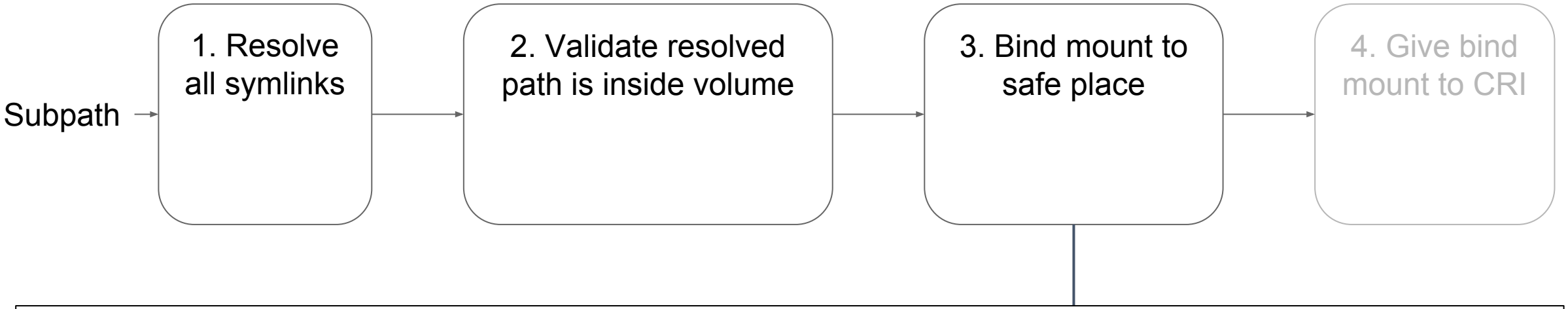
Naive Bind-mount Solution



Naive Bind-mount Solution



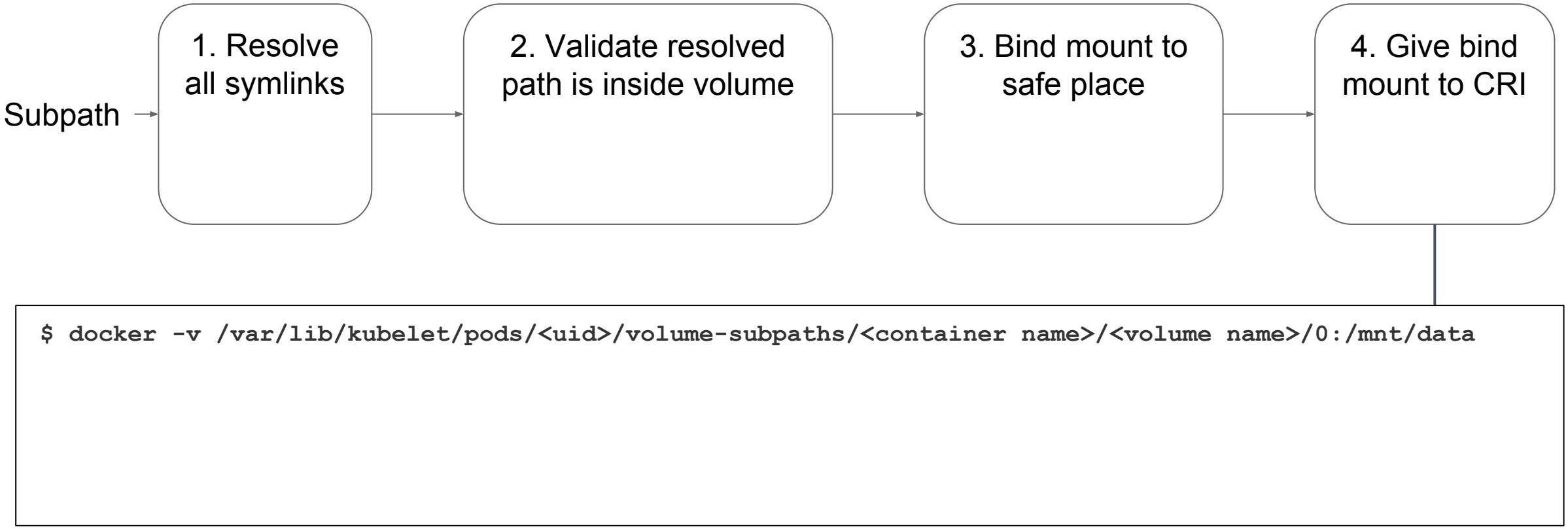
Naive Bind-mount Solution



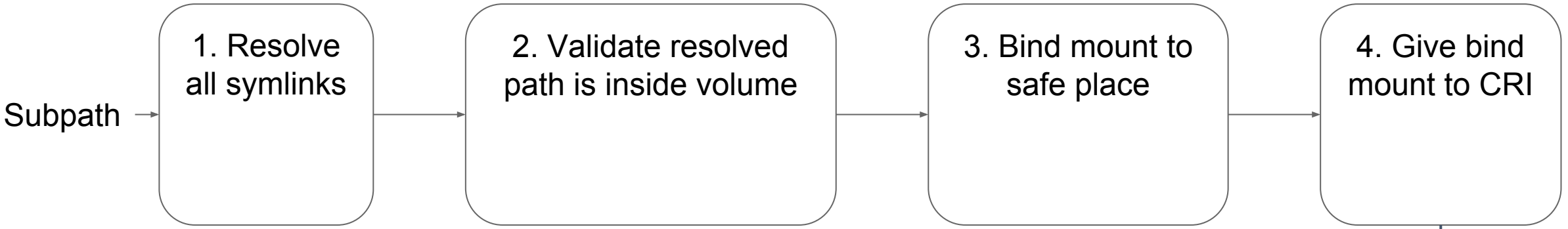

```
$ mount --bind \
    /var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/a/b/c \
    /var/lib/kubelet/pods/<uid>/volume-subpaths/<container name>/<volume name>/0
```

safe place

Naive Bind-mount Solution

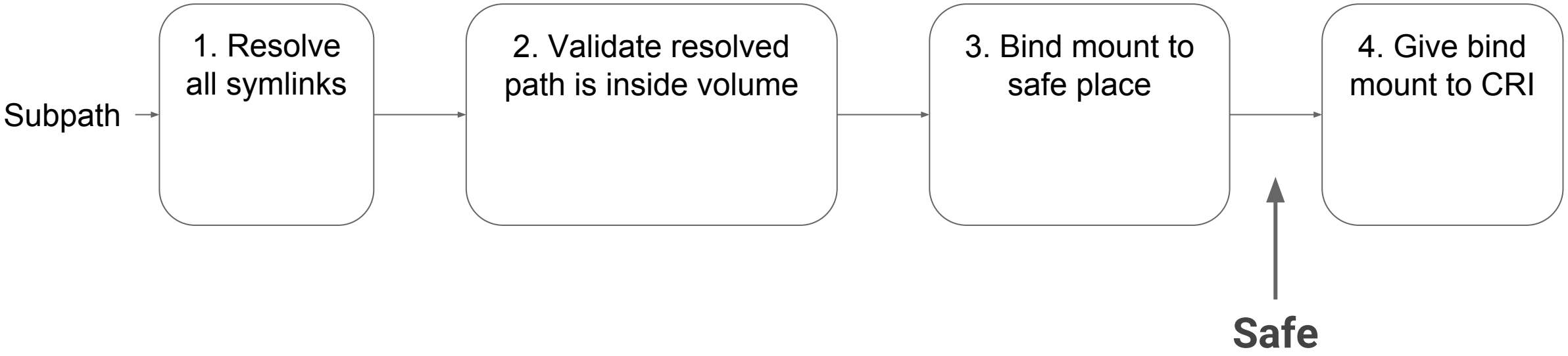


Naive Bind-mount Solution

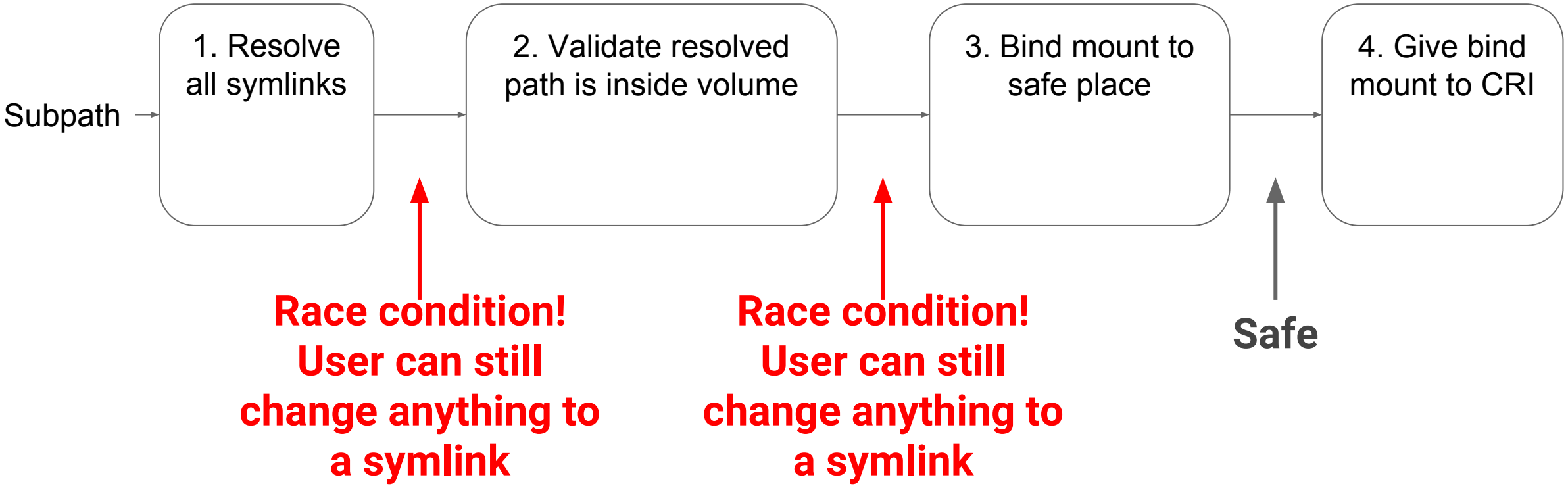



```
$ docker -v /var/lib/kubelet/pods/<uid>/volume-subpaths/<container name>/<volume name>/0:/mnt/data
(was)$   -v /var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1:/mnt/data
```

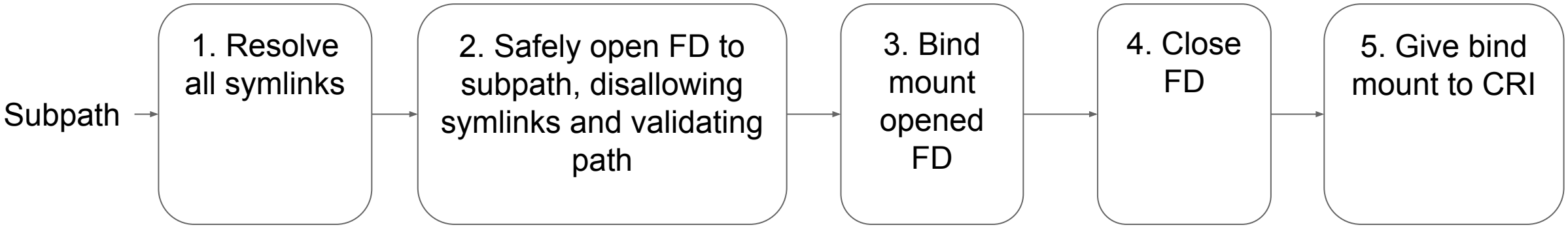
Naive Bind-mount Solution



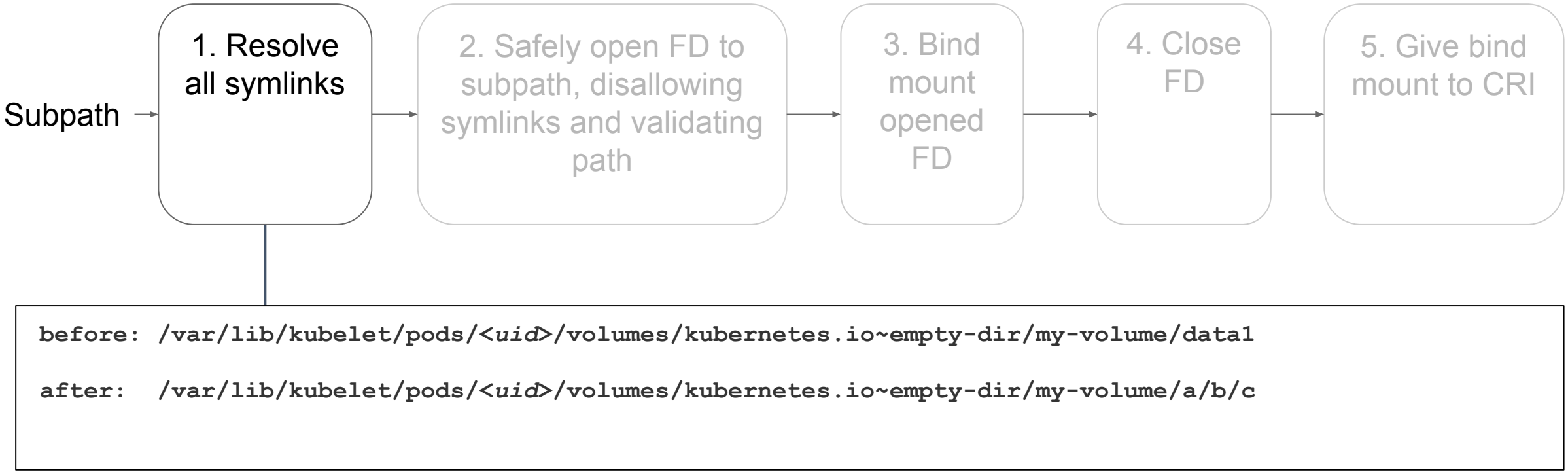
Naive Bind-mount Solution



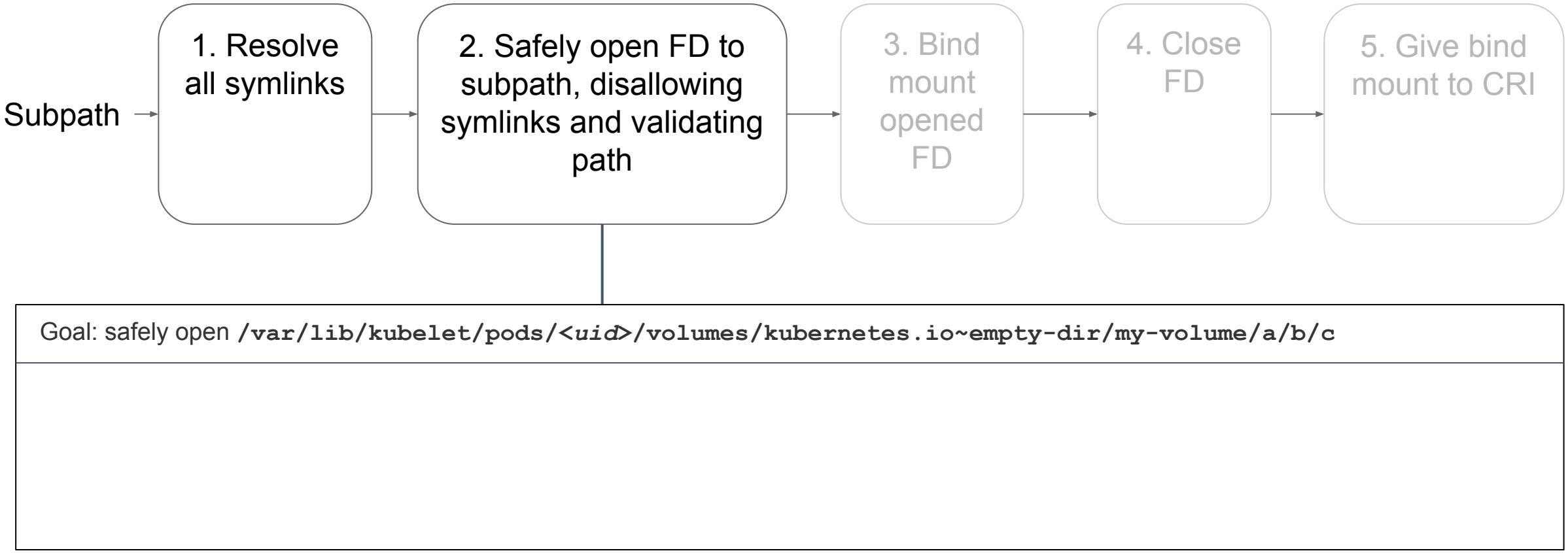
Final Solution



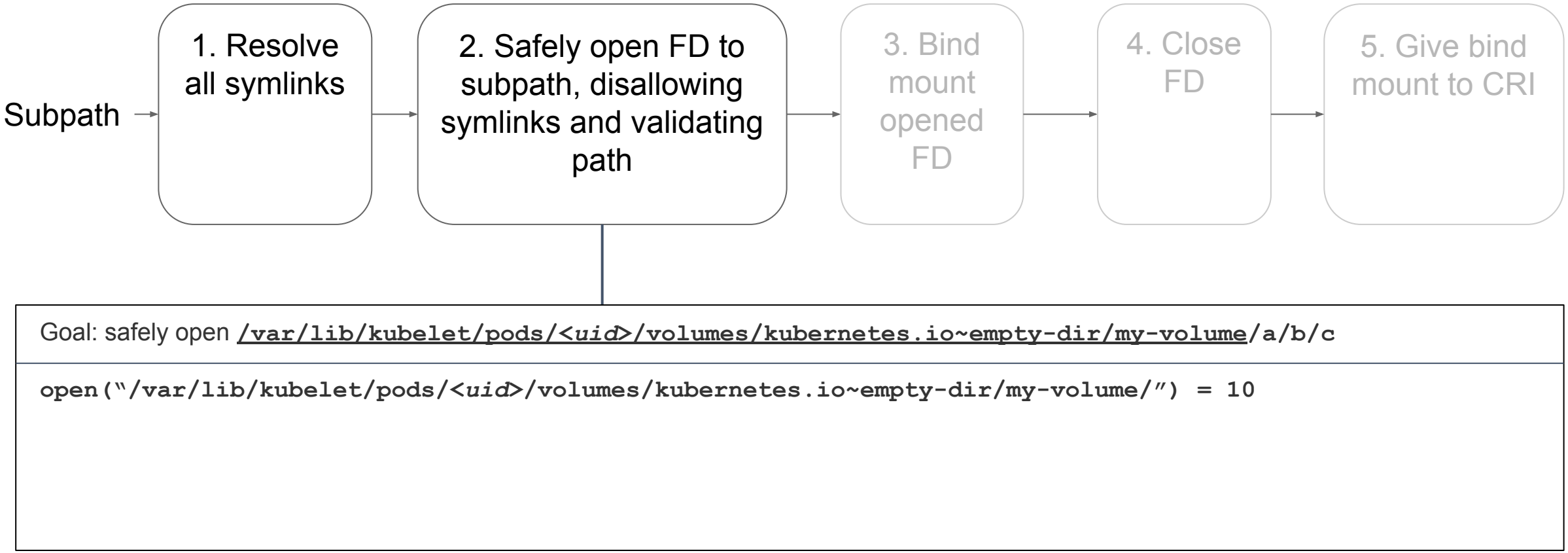
Final Solution



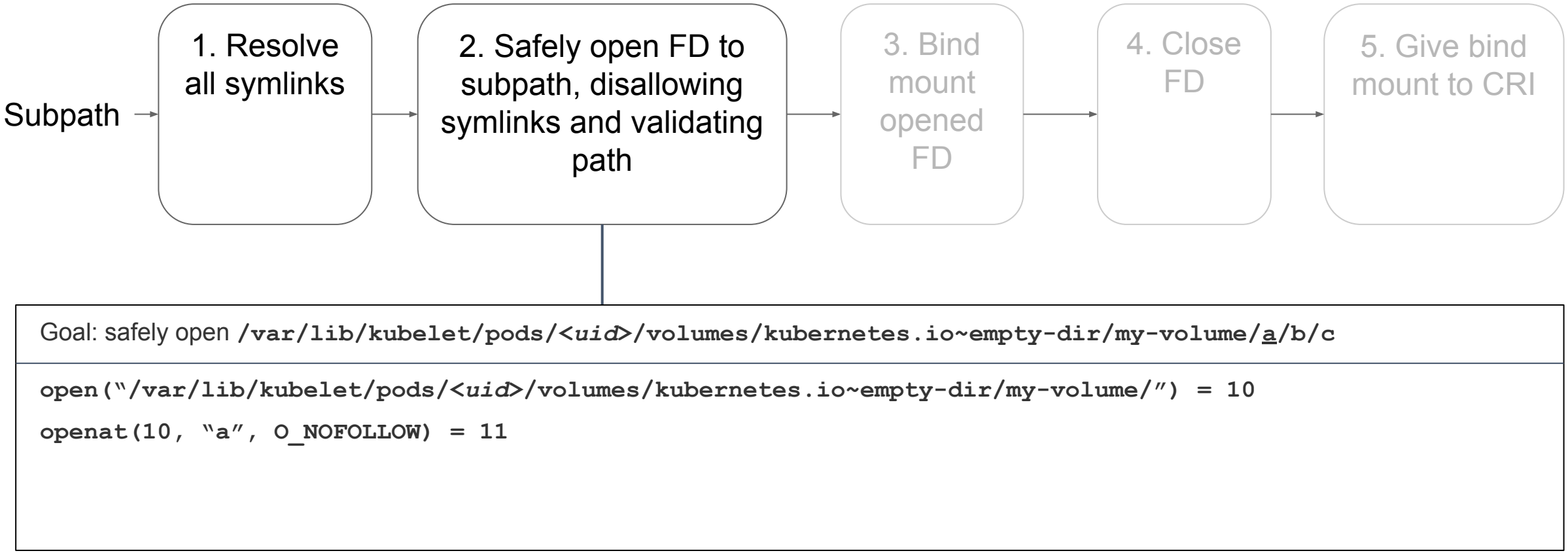
Final Solution



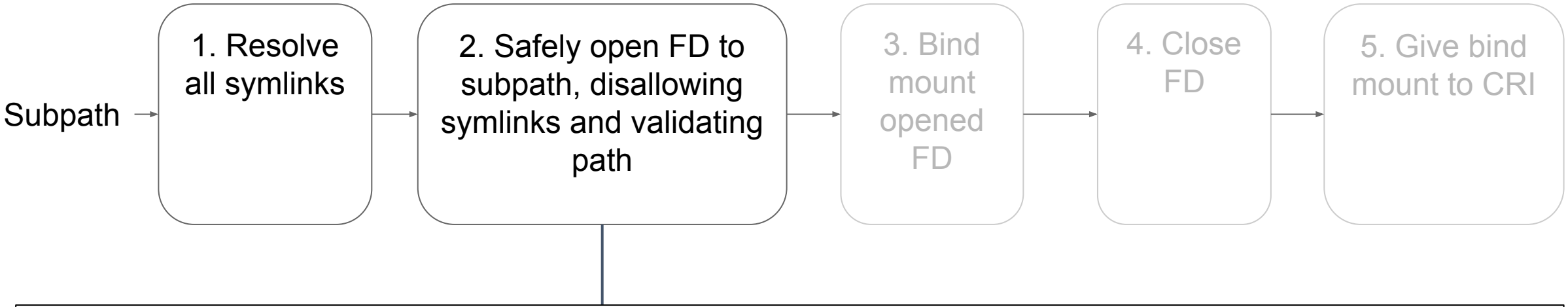
Final Solution



Final Solution



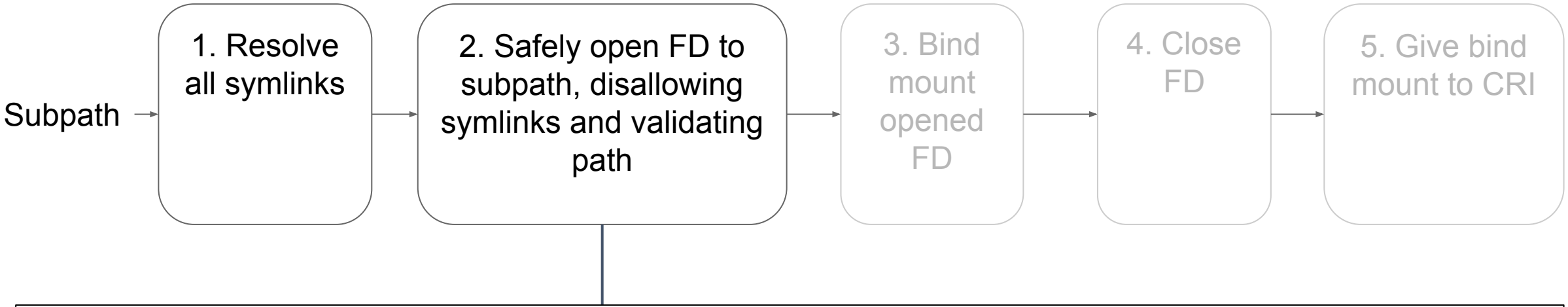
Final Solution



Goal: safely open `/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/a/b/c`

```
open("/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/") = 10
openat(10, "a", O_NOFOLLOW) = 11
openat(11, "b", O_NOFOLLOW) = 12
```

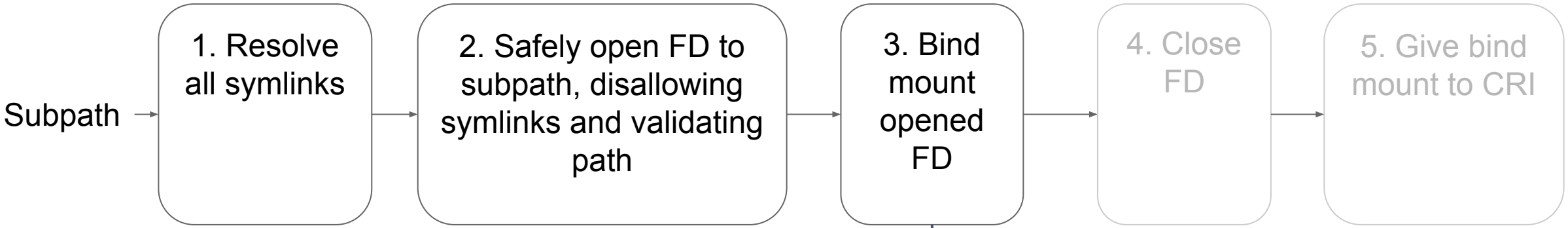
Final Solution



Goal: safely open `/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/a/b/c`

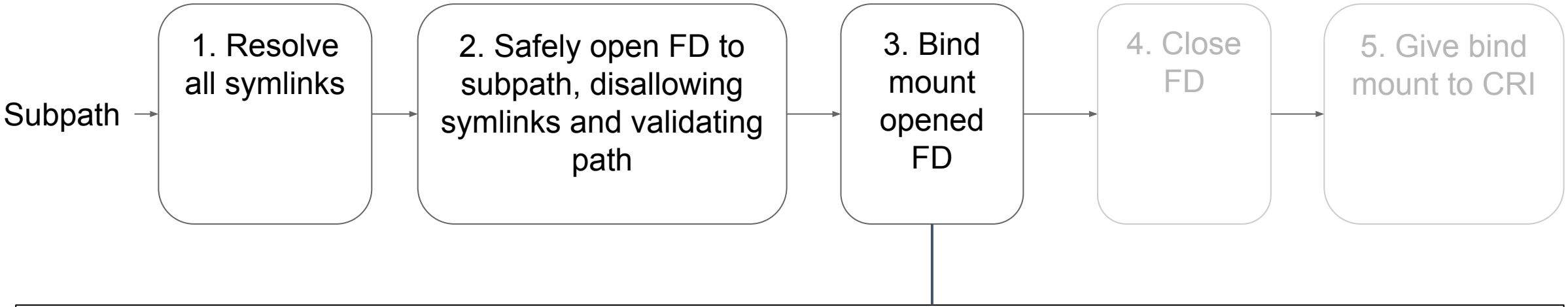
```
open("/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/") = 10
openat(10, "a", O_NOFOLLOW) = 11
openat(11, "b", O_NOFOLLOW) = 12
openat(12, "c", O_NOFOLLOW) = 13
```

Final Solution




```
$ ls -la /proc/<pidof kubelet>/fd/13
13 -> /var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/a/b/c
```

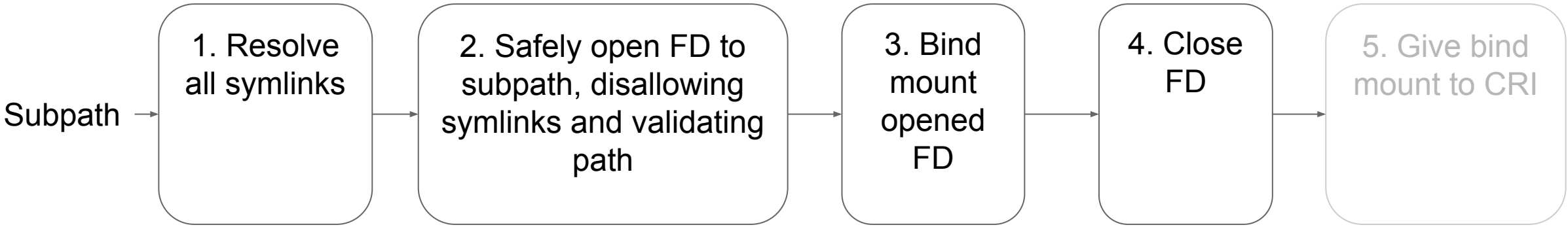
Final Solution



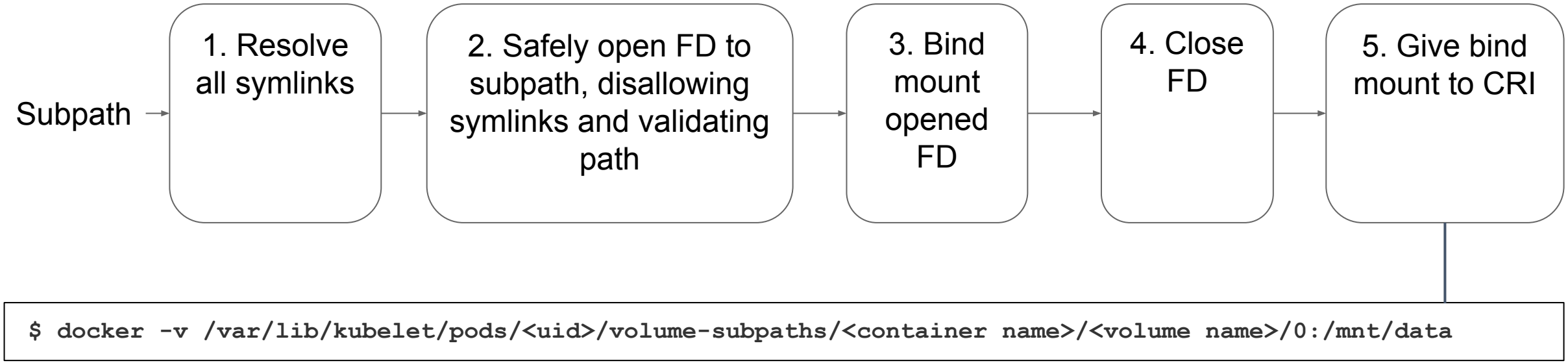

```
$ ls -la /proc/<pidof kubelet>/fd/13
13 -> /var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/a/b/c

$ mount --bind /proc/<pidof kubelet>/fd/13 \
               /var/lib/kubelet/pods/<uid>/volume-subpaths/<container name>/<volume name>/0
```

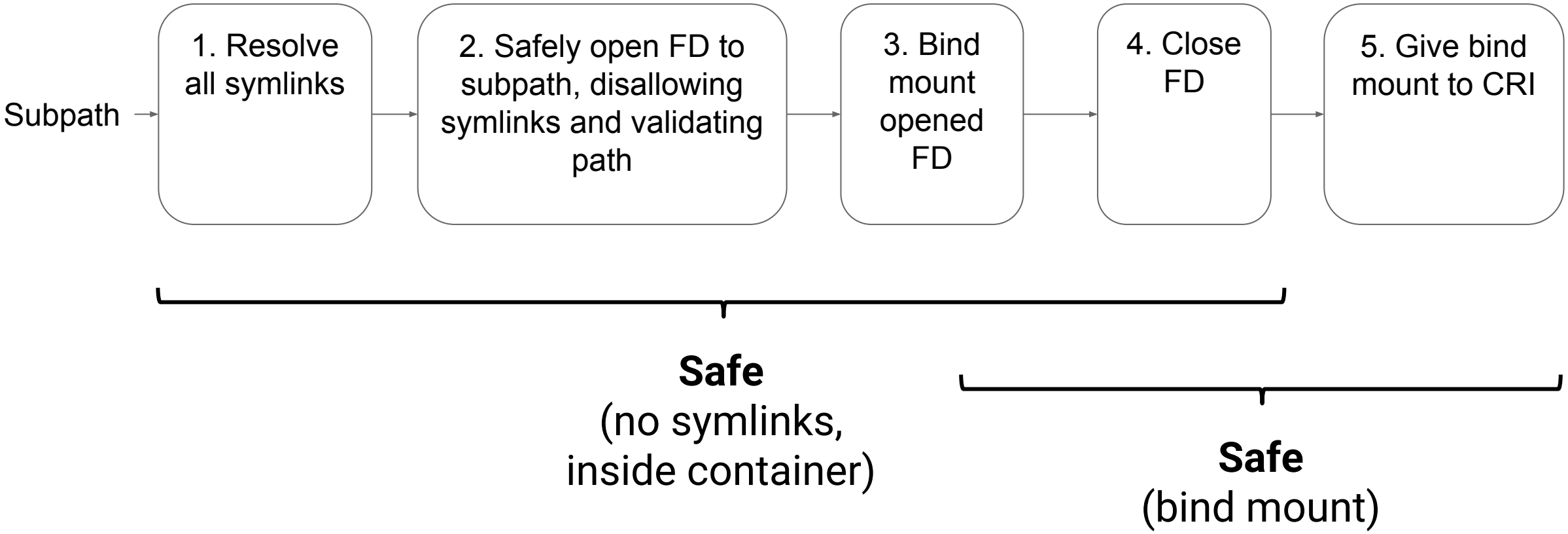
Final Solution



Final Solution



Final Solution



Release Process

Development

github.com/kubernetes/kubernetes-security

- Private, **very** limited access
  - Access revoked once fix released
- Similar process as kubernetes/kubernetes
  - Same test jobs
  - Logs in private buckets
  - Be careful with git pushes
- Coordinated by Product Security Team

Pre-Disclosure

- Coordinated by Product Security Team
- Kubernetes release and branch managers
- 3rd party Kubernetes vendors (“Private Distributors List”)
  - Patch and test under embargo

Public Disclosure

2018-03-12: CVE-2017-1002101 announced

Kubernetes 1.7, 1.8, 1.9 quickly patched and released

Post-mortem document

Securing Volumes

Secure Practices Now

Don’t run containers as root user

- Use PodSecurityPolicy to require pods run as non-root
- **Caveat**: containers still run as root gid
  - K8s 1.10: RunAsGroup alpha feature

Use PodSecurityPolicy to restrict volume access

- Whitelist allowed volume types
- Restrict **both** allowed HostPath prefixes and readOnly (K8s 1.11)

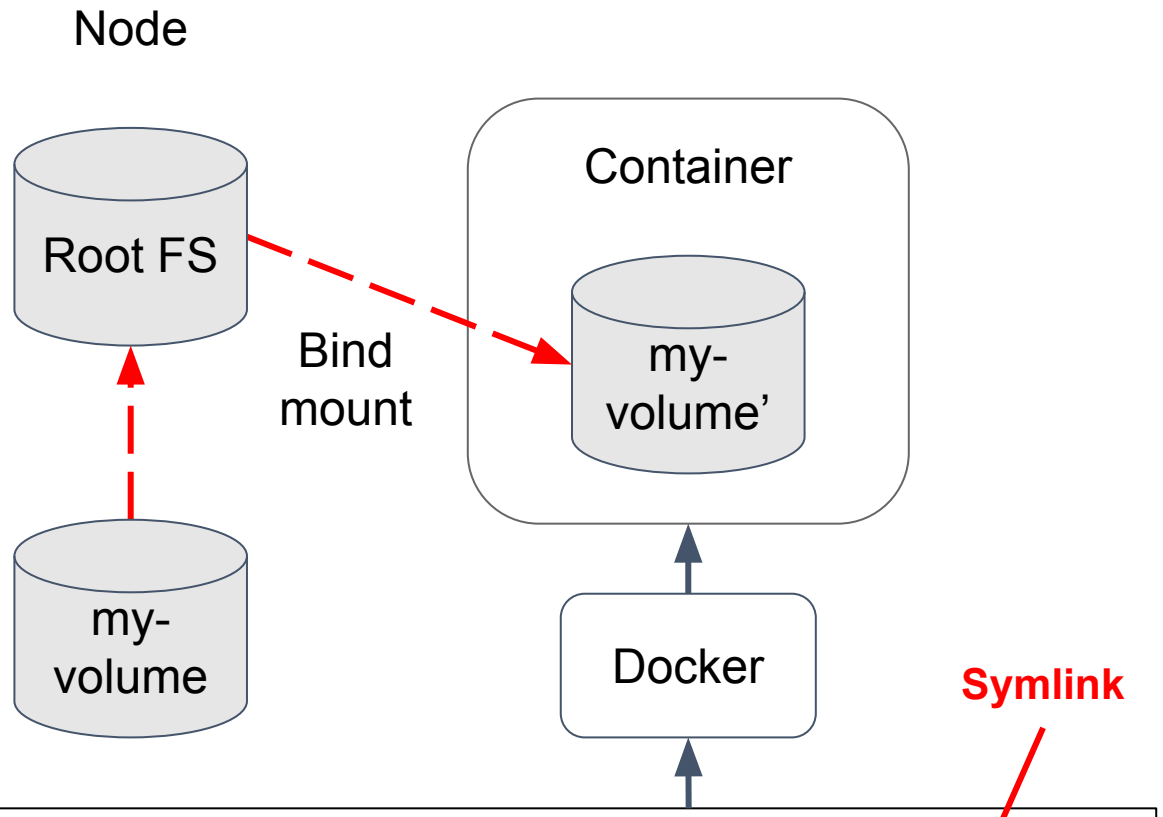
Future Improvements

Core Principle

Multiple security boundaries around untrusted code

Vulnerability


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```

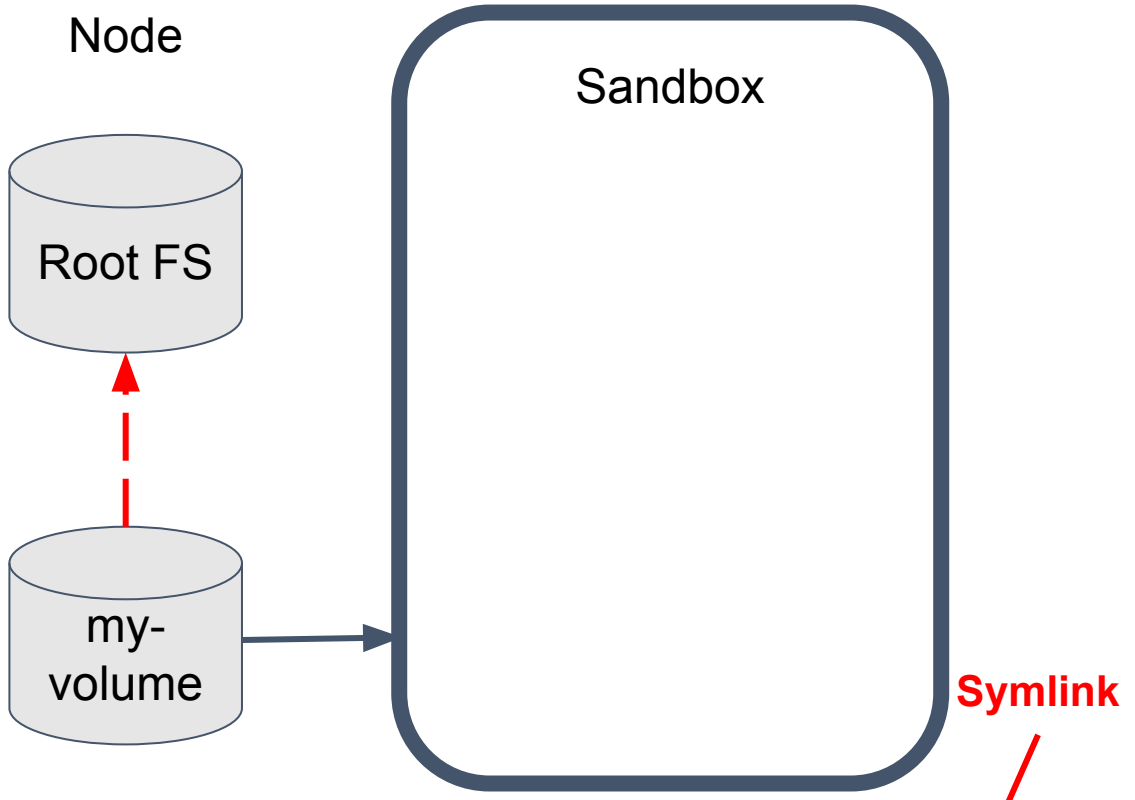


Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Future: Sandboxed Subpath?


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```

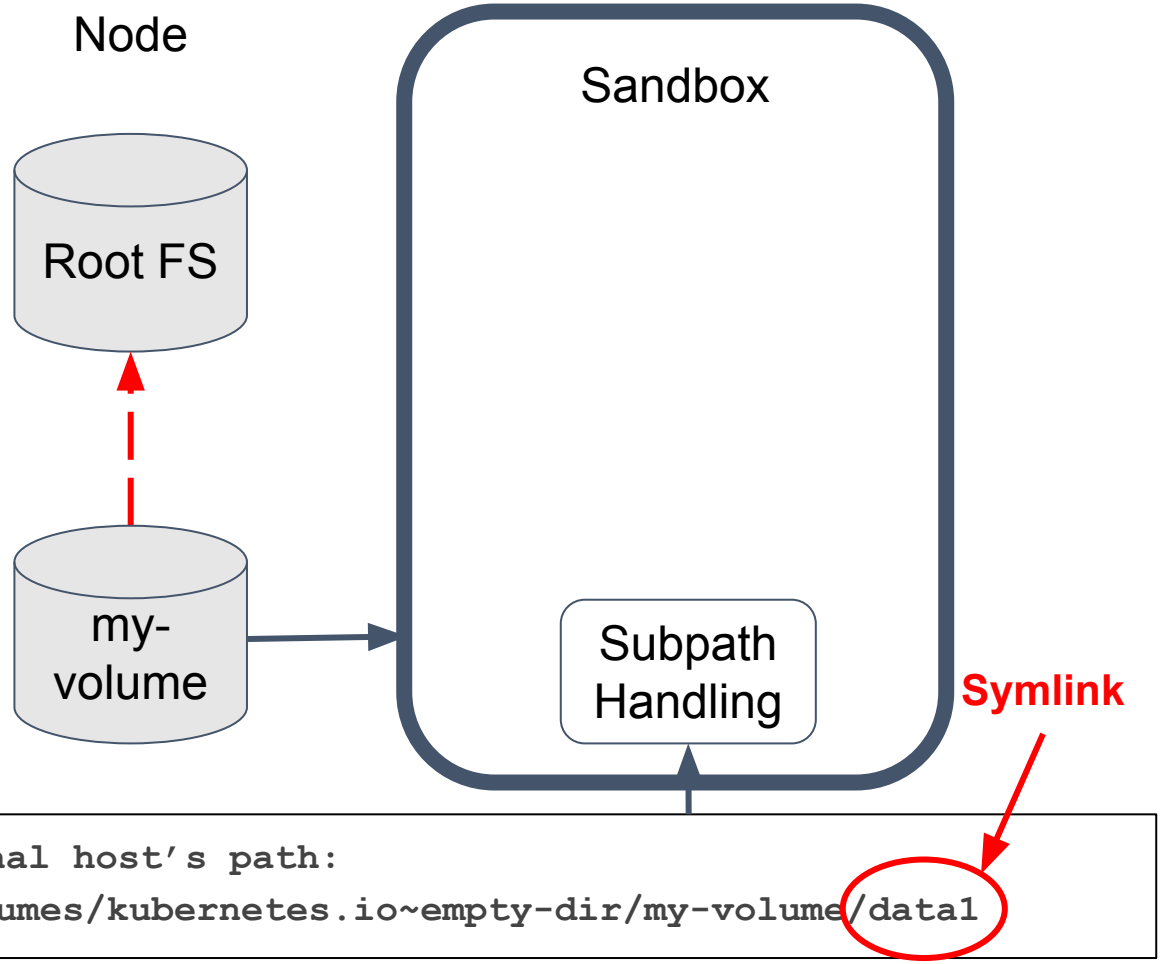


Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Future: Sandboxed Subpath?


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```

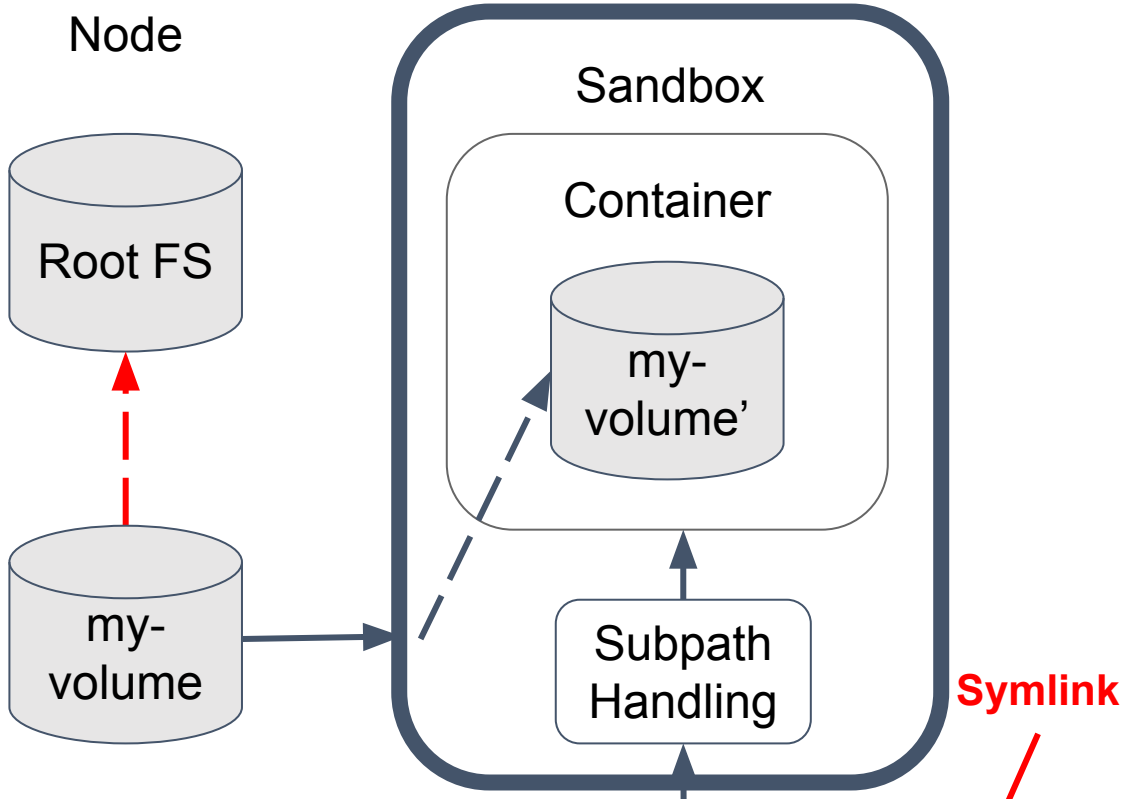


Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Future: Sandboxed Subpath?


```
kind: Pod
spec:
  containers:
  - name: my-container
    volumeMounts:
    - name: my-volume
      mountPath: /mnt/data
      subPath: data1
  volumes:
  - name: my-volume
    emptyDir: {}
```



Final host's path:

/var/lib/kubelet/pods/<uid>/volumes/kubernetes.io~empty-dir/my-volume/data1

Summary

Follow the Kubernetes security disclosure process

Be extra cautious when handling untrusted paths

- symlink race
- time of check to time of use

Set restrictive policies and use multiple security boundaries

Acknowledgments

Thanks to Maxim Ivanov for reporting the vulnerability!

Get Involved!

Kubernetes product security team

Secure Container Isolation

- Overall: sig-node
- Volumes: sig-storage
- “Recent Advancements in Container Isolation” - today 1:45pm


KubeCon
CloudNativeCon
North America 2018