



Lars Lefebvre

Kapers op de K8s kust

#whoami

BE België



Product Security@ING



Bassist



Padel- en tennis speler



NBA fan



Smashed burgers



ANNA CON
OX7E8

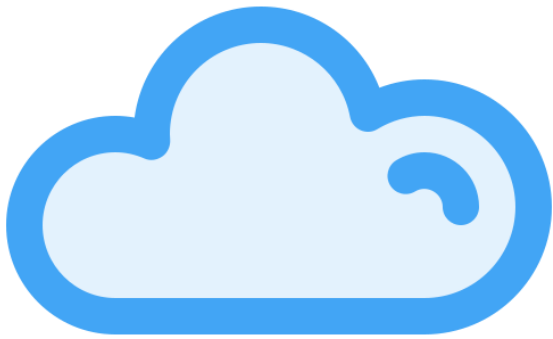


Google



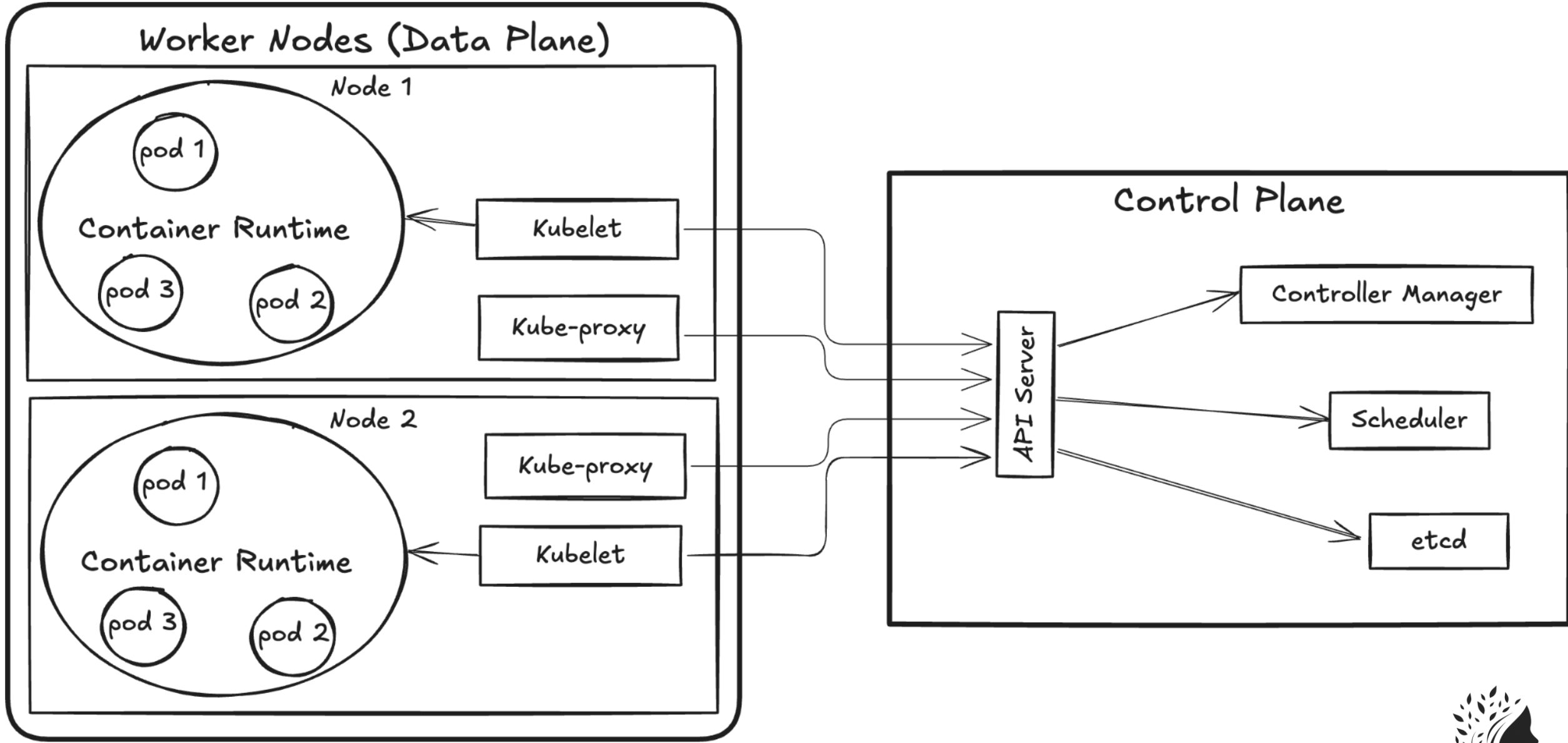
ANNA CON
OX7E8

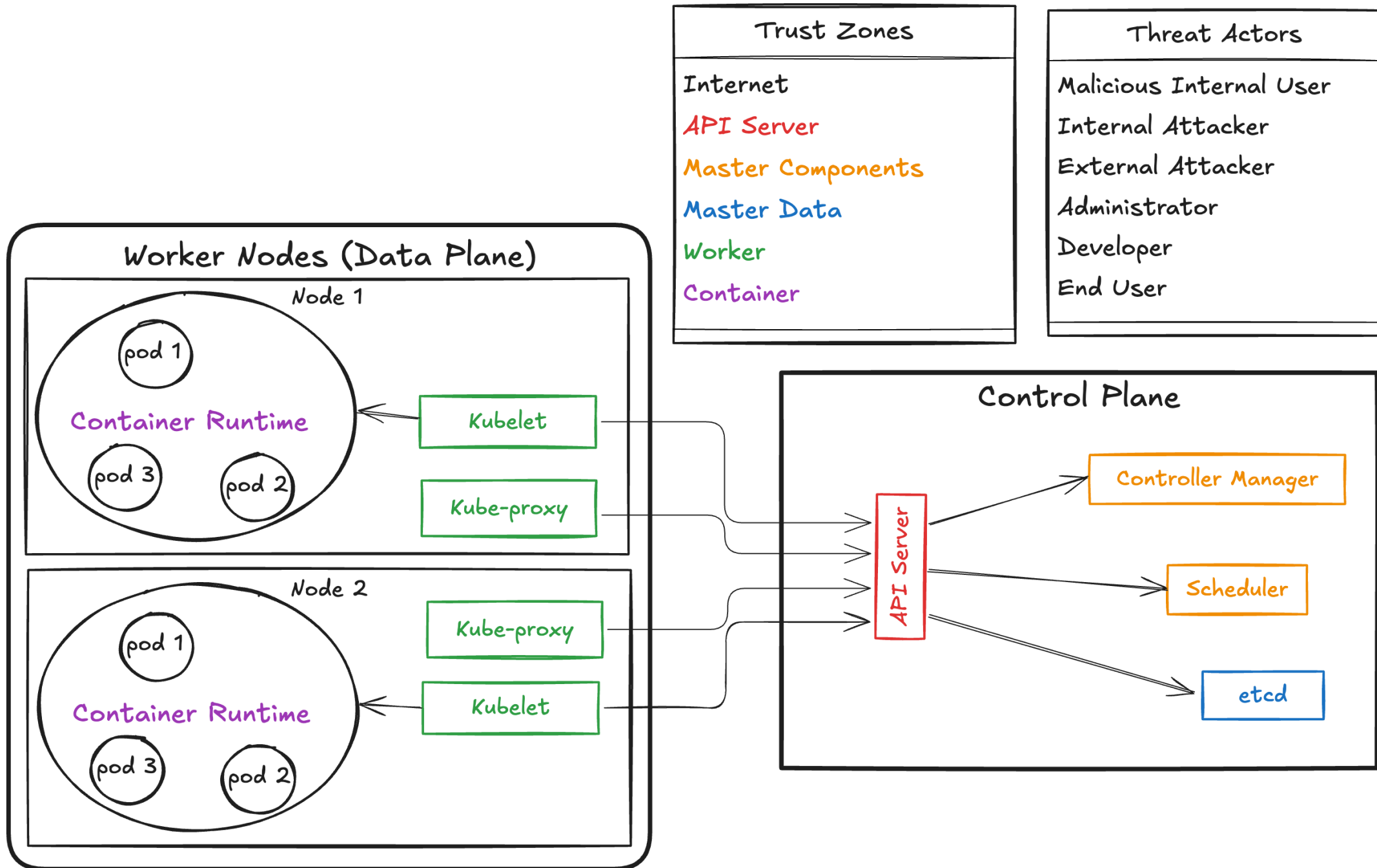
4 C's of Cloud-Native Security

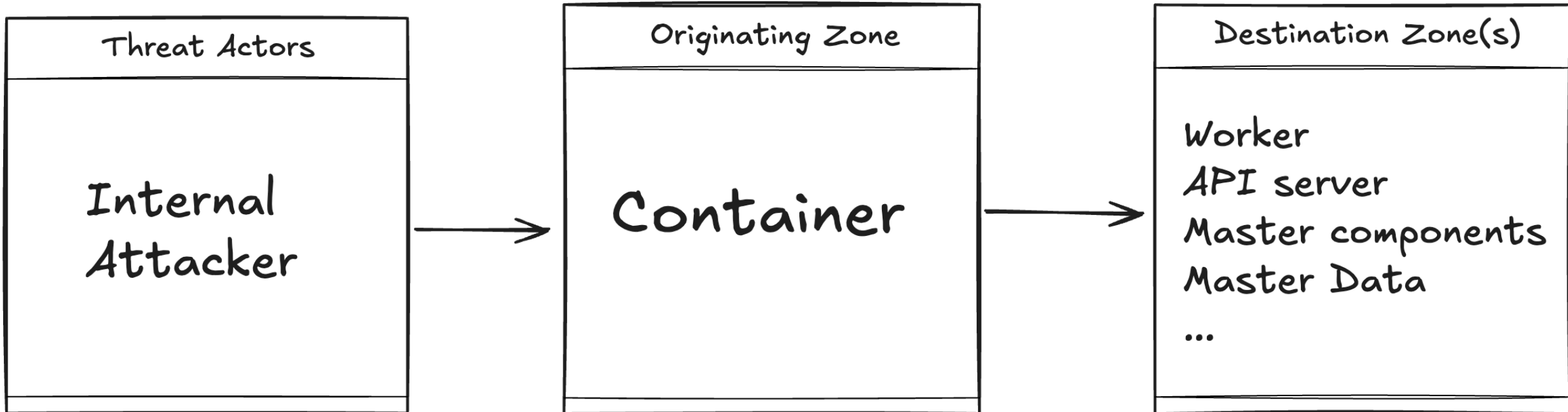


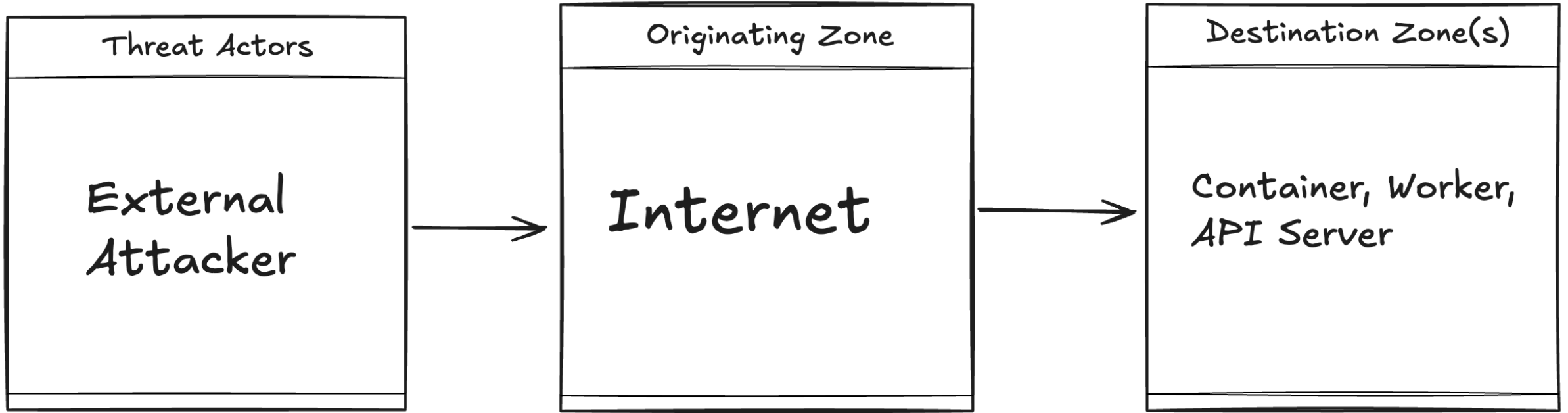
Kubernetes (Security) Concepts







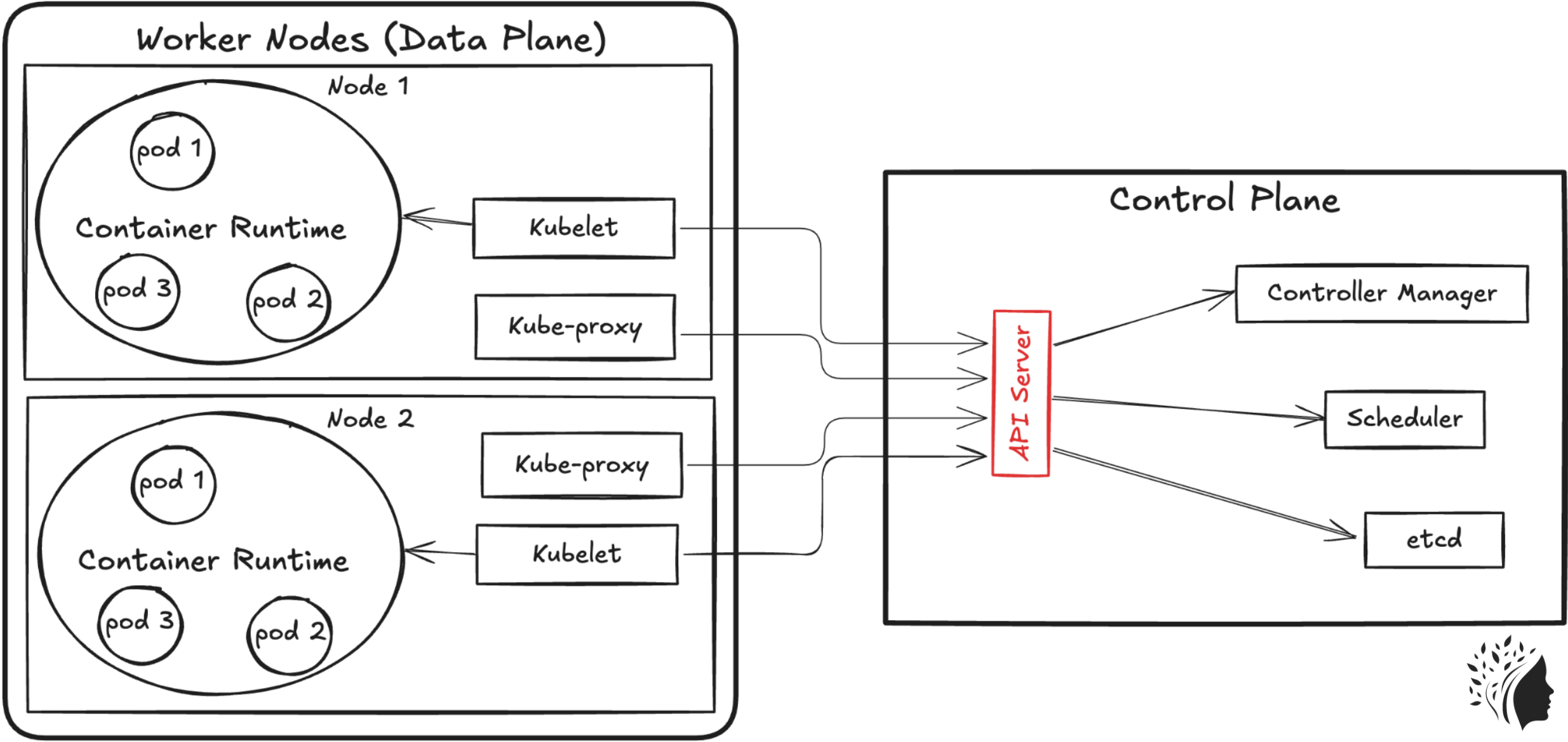




Initial access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access Kubernetes API server	Access cloud resources	Images from a private registry	Data destruction
Compromised image In registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Collecting data from pod	Resource hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from proxy server	Application credentials in configuration files	Exposed sensitive interfaces	Application credentials in configuration files		
Exposed sensitive interfaces	SSH server running inside container	Container service account			Access managed identity credentials	Instance Metadata API	Writable hostPath mount		
	Sidecar injection	Static pods			Malicious admission controller		CoreDNS poisoning		
							ARP poisoning and IP spoofing		



Overly Permissive RBAC



Kubernetes RBAC

Namespace

Roles

RoleBindings

Define permissions

Associate roles with users, groups or service accounts

Cluster

ClusterRoles

ClusterRoleBindings

Define cluster-wide permissions

Associate cluster roles with users, groups or service accounts





```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: demo-ns
  name: demo-role
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["create", "get", "watch", "list", "exec", "patch"]
- apiGroups: [""]
  resources: ["pods/exec"]
  verbs: ["create"]
```



```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: demo-role-binding
  namespace: demo-ns
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: demo-role
subjects:
- kind: ServiceAccount
  name: demo-sa
  namespace: demo-ns
```



```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: demo-cluster-role
rules:
- apiGroups: [""]
  resources: ["nodes", "pods", "pods/exec"]
  verbs: ["get", "watch", "list", "exec", "create"]
```



```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: demo-cluster-role-binding
subjects:
- kind: ServiceAccount
  name: demo-sa
  namespace: demo-ns
roleRef:
  kind: ClusterRole
  name: demo-cluster-role
  apiGroup: rbac.authorization.k8s.io
```



cluster-admin
binding





larslefevre@MacBook-Pro-van-Lars K03-RBAC %

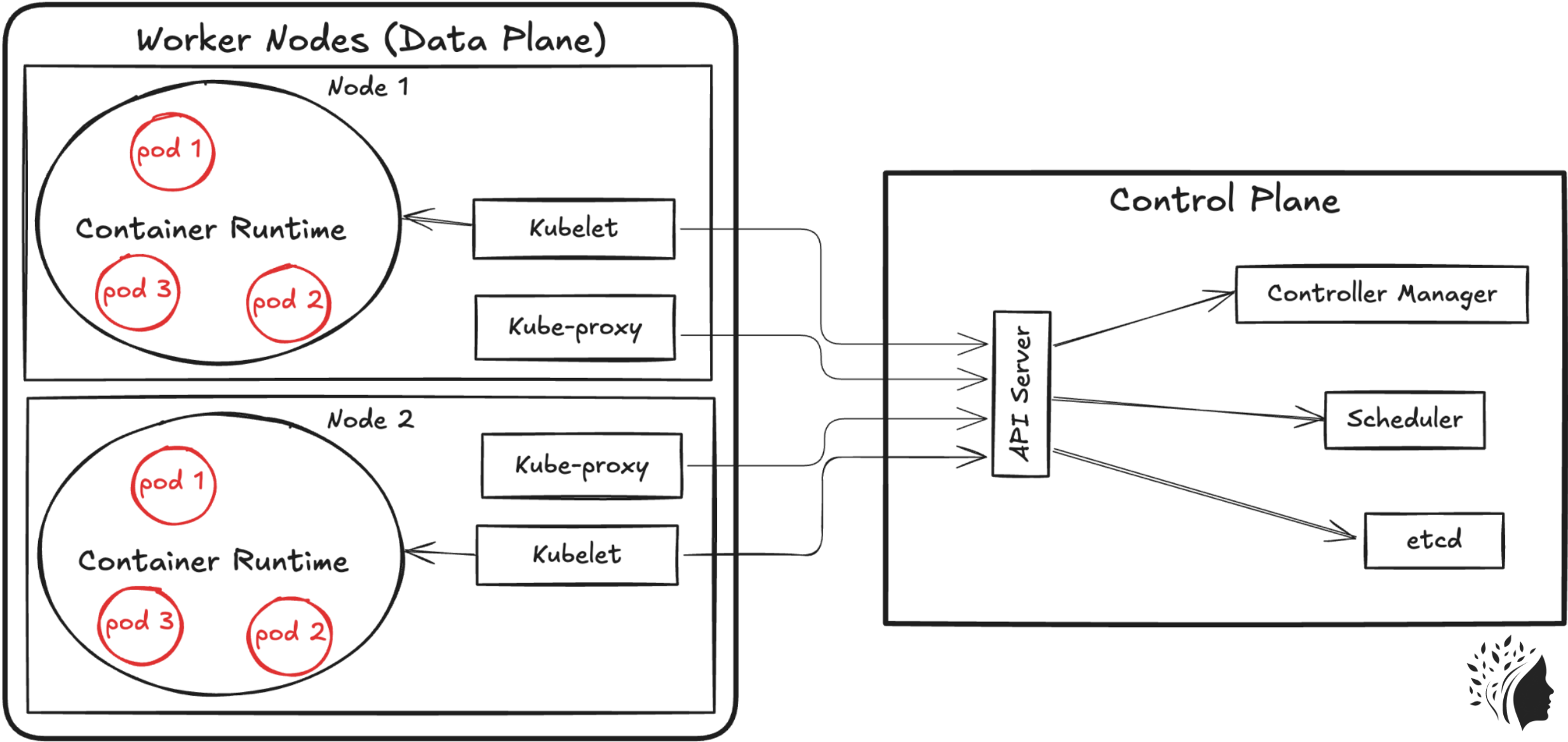


Avoiding Overly Permissive RBAC Configurations



ANIACON
OX7E8

Insecure Workload Configurations





```
apiVersion: v1
kind: Pod
metadata:
  name: demo-app-insecure-config
spec:
  securityContext:
    runAsNonRoot: false
    runAsUser: 0
```



```
apiVersion: v1
kind: Pod
....
spec:
  containers:
  - name: demo-app-insecure-config
    image: larslefebvre/h4cking-workshop-demo-app:867361
    imagePullPolicy: IfNotPresent
    securityContext:
      privileged: true
    ...
  serviceAccountName: demo-app-insecure-config
  ...
```





```
apiVersion: v1
kind: Pod
....
spec:
  hostNetwork: true
  hostPID: true
  hostIPC: true
  containers:
  - name: demo-app-insecure-config
    ...
    volumeMounts:
    - mountPath: /host
      name: noderoot
  serviceAccountName: demo-app-insecure-config
  volumes:
  - name: noderoot
    hostPath:
      path: /
```





```
apiVersion: v1
kind: Pod
....
spec:
  hostNetwork: true
  hostPID: true
  hostIPC: true
  containers:
  - name: demo-app-insecure-config
    ...
    volumeMounts:
    - mountPath: /host
      name: noderoot
  serviceAccountName: demo-app-insecure-config
  volumes:
  - name: noderoot
    hostPath:
      path: /
```





```
apiVersion: v1
kind: Pod
....
spec:
  containers:
  - name: demo-app-insecure-config
    ....
    securityContext:
      capabilities:
        drop:
          - ALL
        add: ["NET_ADMIN", "SYS_TIME", "SYS_ADMIN"]
    ....
```

privileged
container



hostPath mount



```
larslefebvre@MacBook-Pro-van-Lars K01-InsecureConfig %
```

Remedies for Insecure Workload Configuration

Deze partners hebben een ❤️ voor ANNACON 0x7E8.



ANNACON
0x7E8

