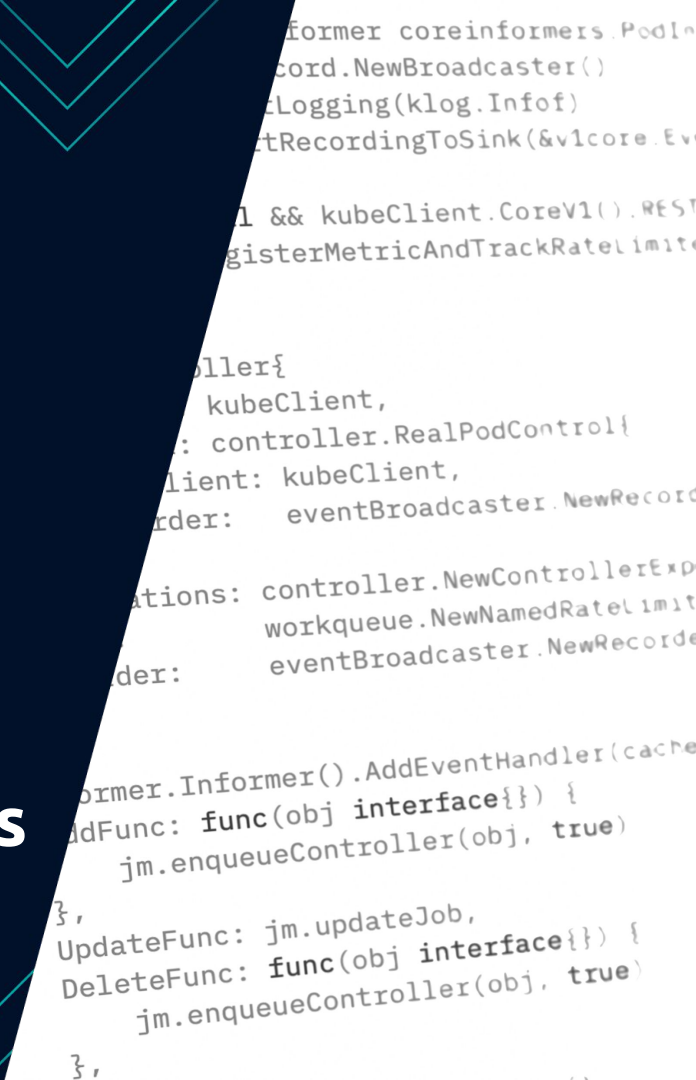
KUBERMATIC

Container Day Security, Mar 8th 2023

Lines of Defence:
Securing Your Kubernetes Clusters



Koray Oksay
Kubernetes Consultant

koray@kubermatic.com
@korayoksay
koksay

Who am I?

- Consultant / SRE / Instructor @Kubermatic
- Working remotely from Istanbul
- Interested in Linux, Kubernetes, and cloud technologies
- CKA | CKAD | CKS | RHCE
- Linux Foundation Trainer for CK.*

Cryptojacking and Crypto Mining – Tesla, Kubernetes, and Jenkins Exploits

NEWS

Unprotected Kubernetes consoles expose firms to cryptojacking

A number of big companies have been targeted by crytojacking attacks, where cyber criminals hijack computing power to mine cryptocurrencies, but some have unprotected Kubernetes consoles in common

mine cryptocurrency

Updated: Researchers have discovered that Tesla's AWS cloud systems were compromised for the purpose o
cryptojacking.

Written by **Charlie Osborne**, Contributor
Posted in Zero Day on February 20, 2018 | Topic: Security

RELATED

The 5 best free music
Top streaming apps a
picks

Get 1TB of cloud stora
lifetime for just $140

s Cryptojacking Warning

owing the news that Tesla's cloud

Related to This Story

Why Enclaves are Taking over the Security World

Reduce Security Complexity for Cloud Architectures

ATEST VICTI
JACKING AT
COULD COME SOON

It's the latest in a long line of attacks.

Tesla has fallen victim to a cryp
using visitors' computer resour

The issue, as discovered by the

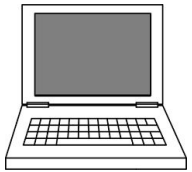
https://www.shodan.io/search?query=kubernetes

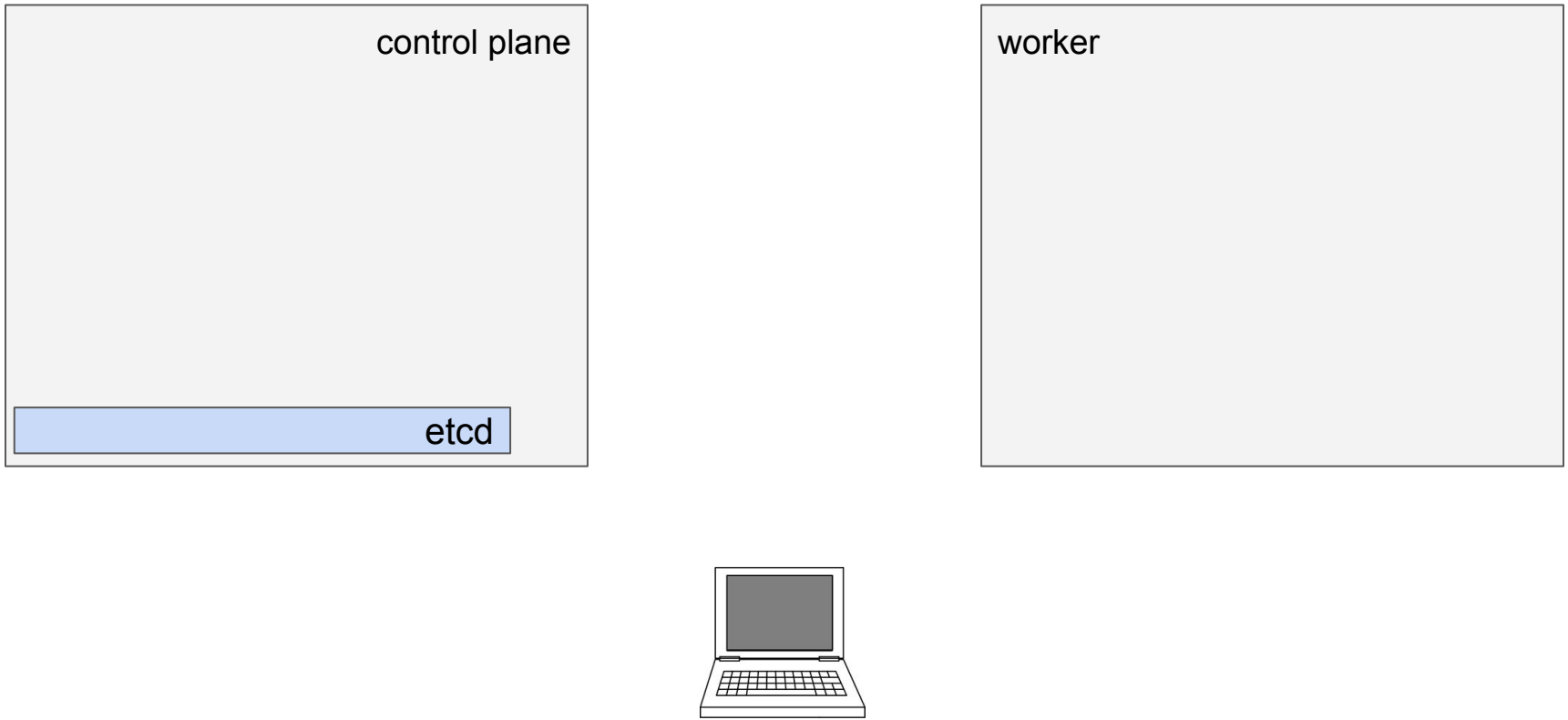
KUBERMATIC

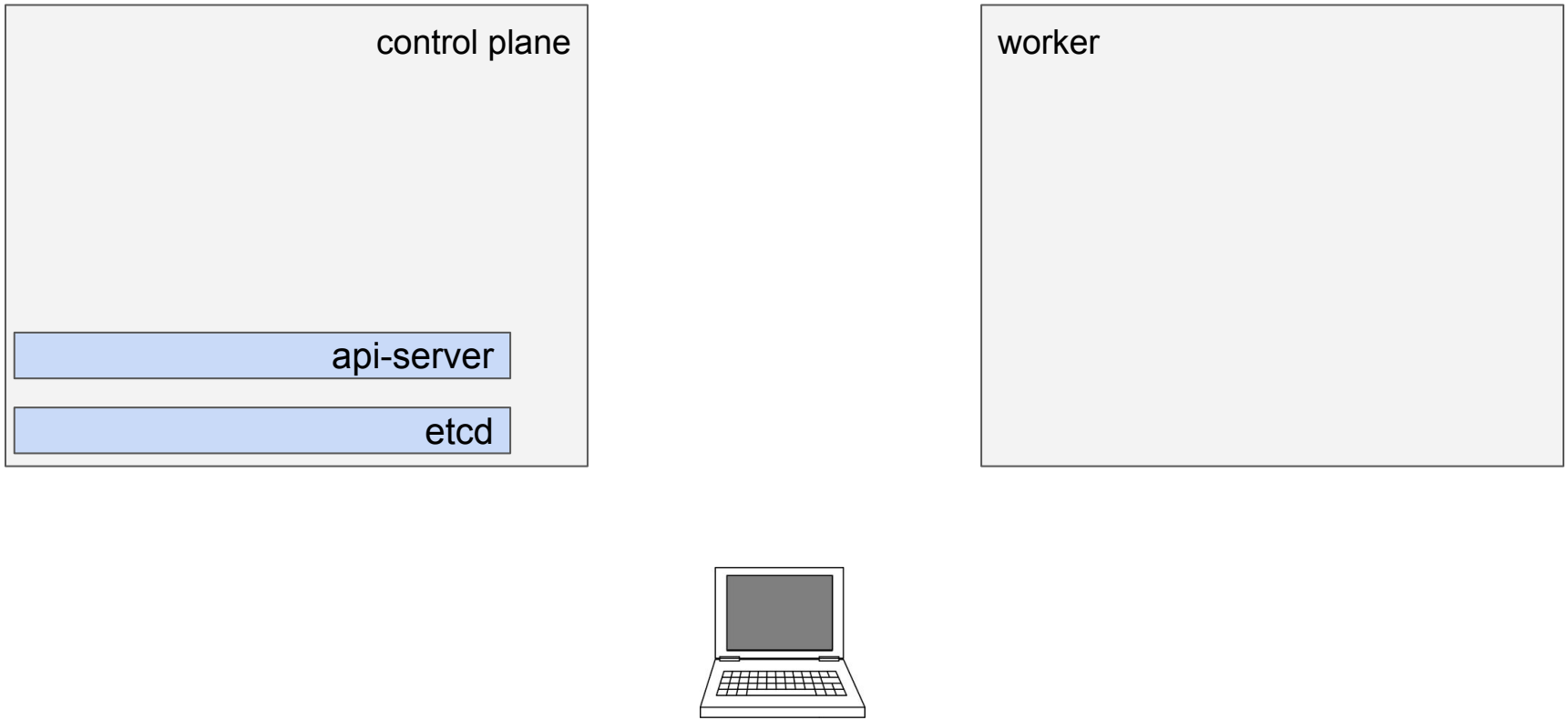
Kubernetes Architecture

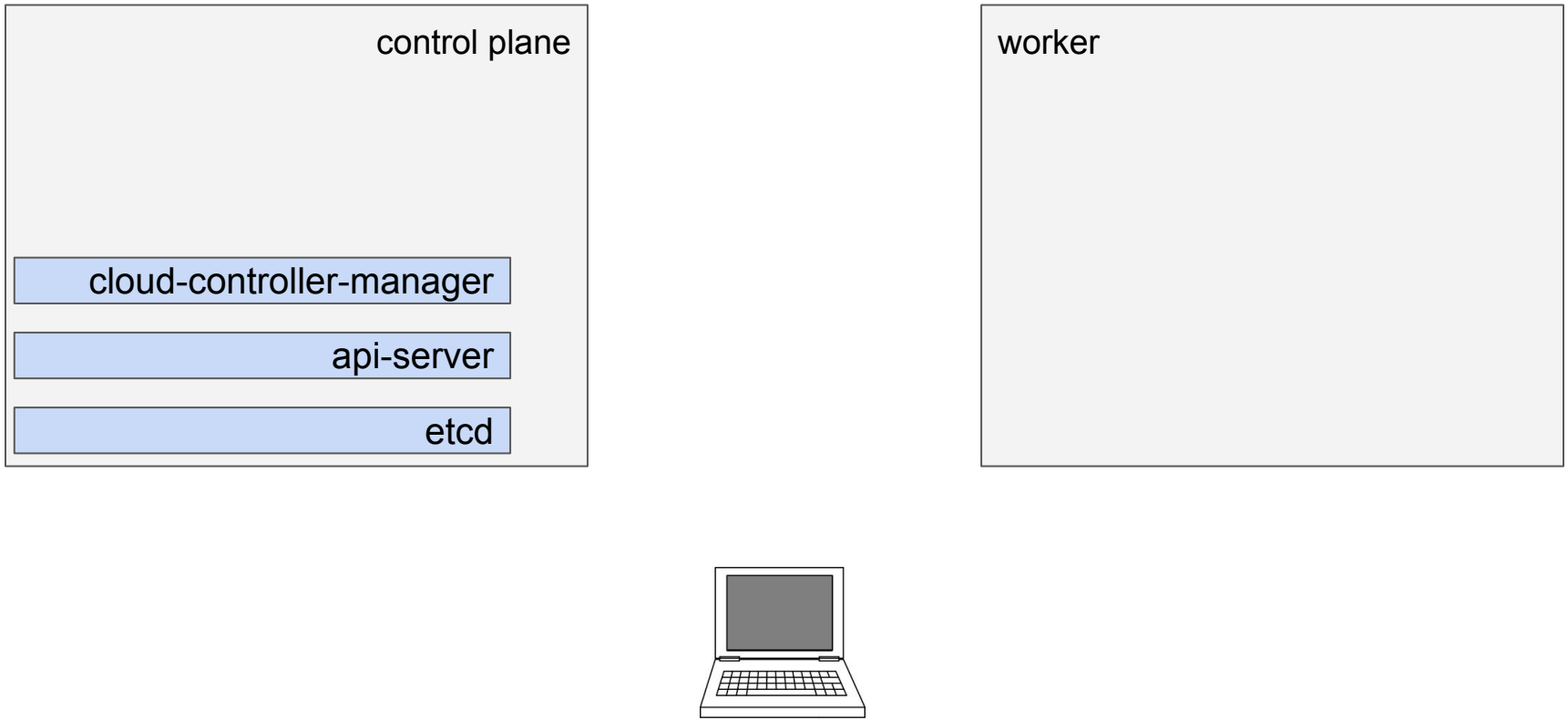
control plane

worker




control plane
etcd
worker


control plane
api-server
etcd
worker


control plane
cloud-controller-manager
api-server
etcd
worker

control plane

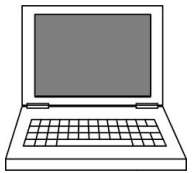
scheduler

cloud-controller-manager

api-server

etcd

worker



control plane

controller-manager

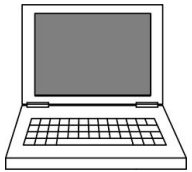
scheduler

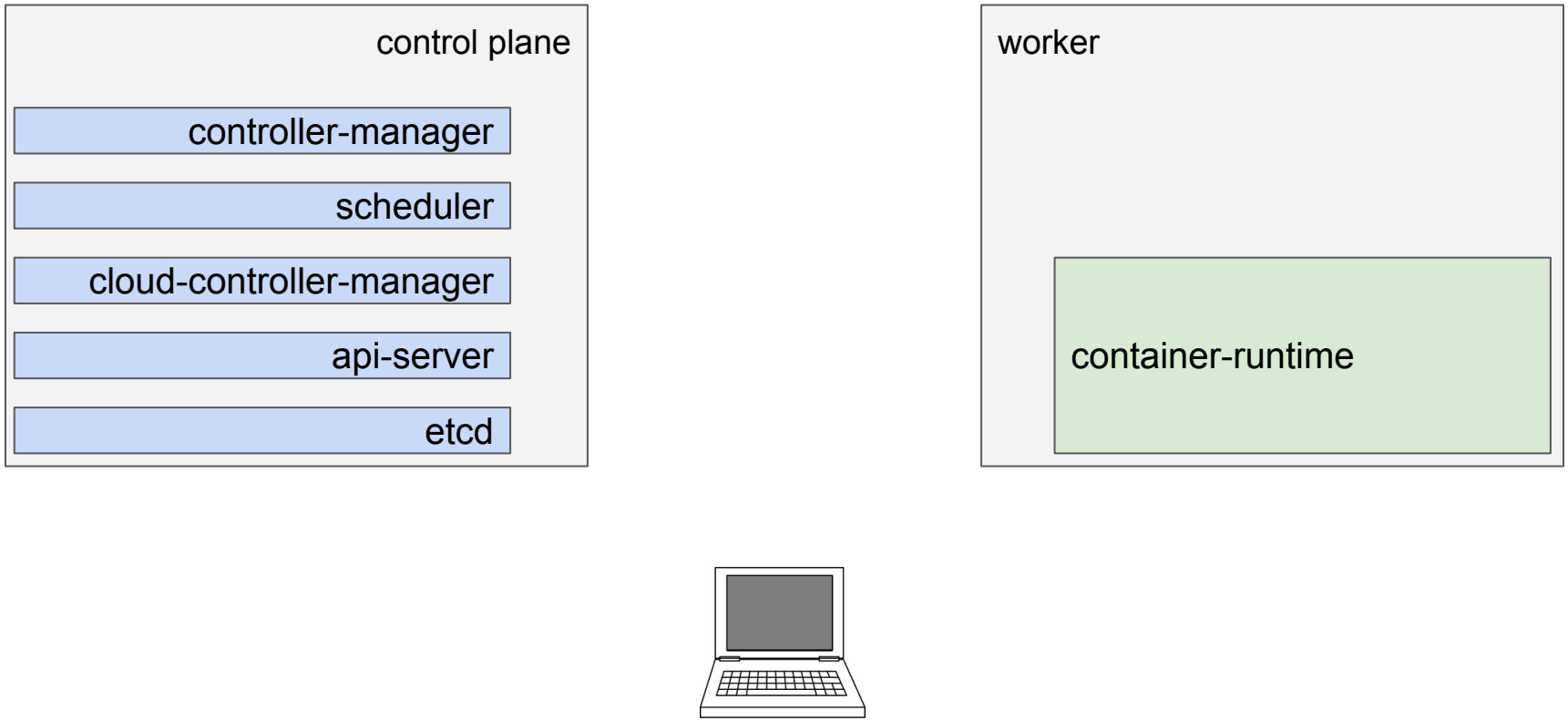
cloud-controller-manager

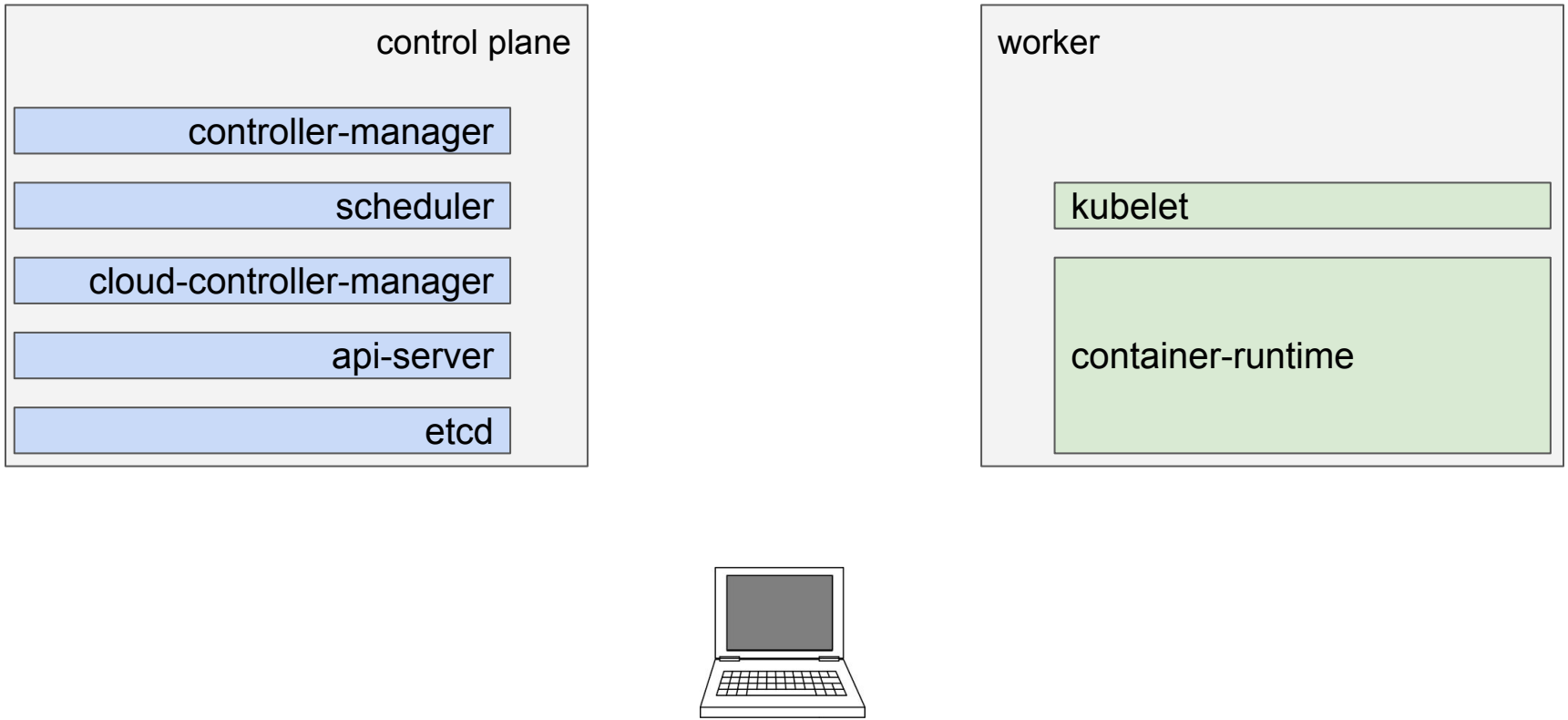
api-server

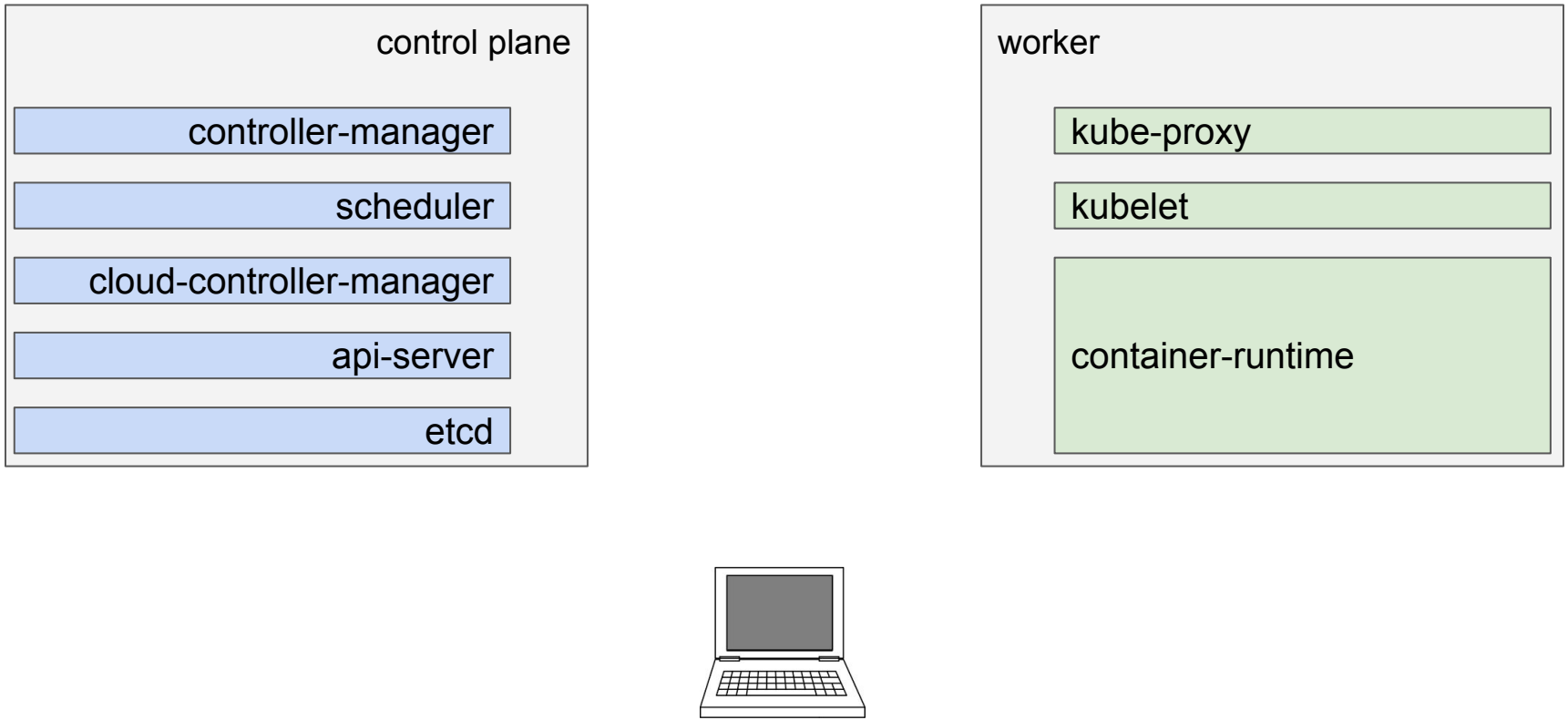
etcd

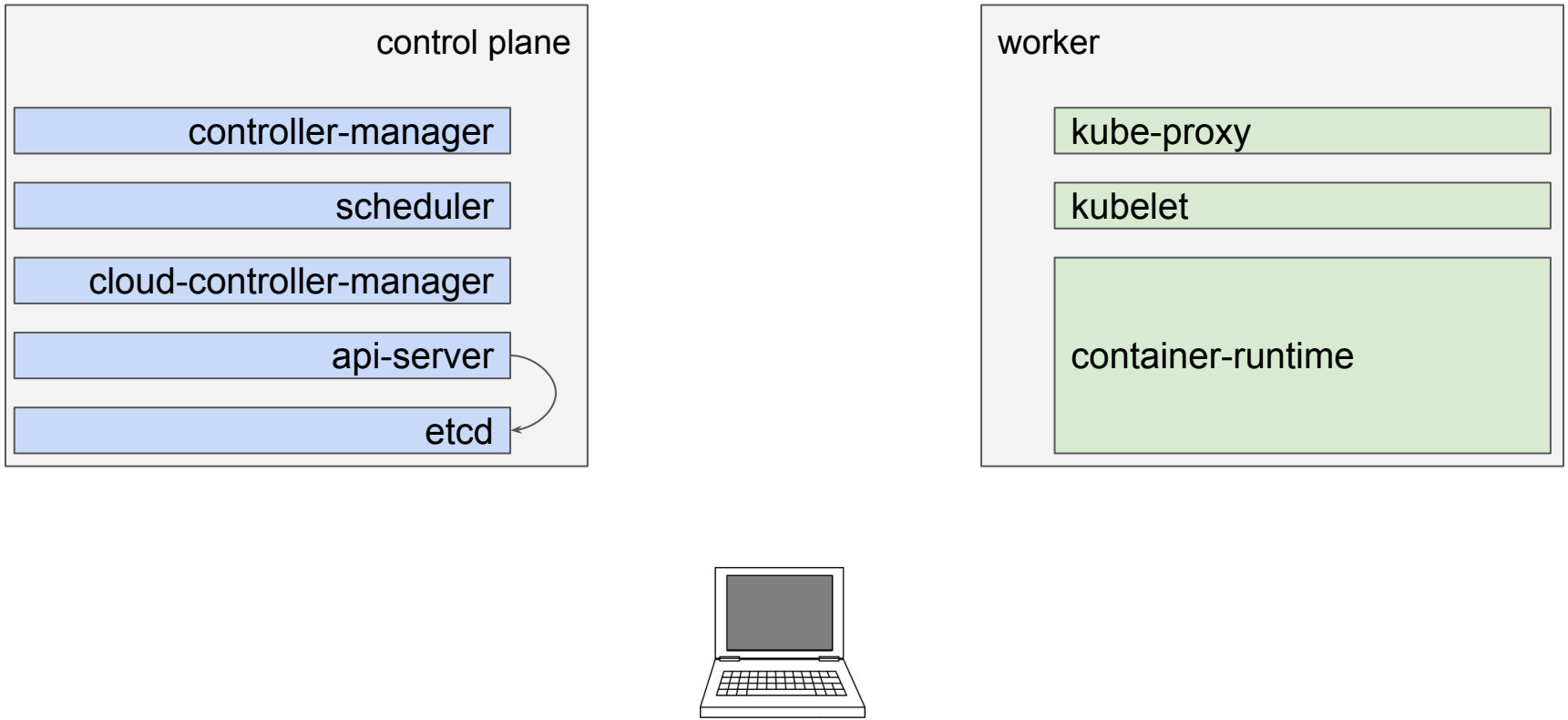
worker

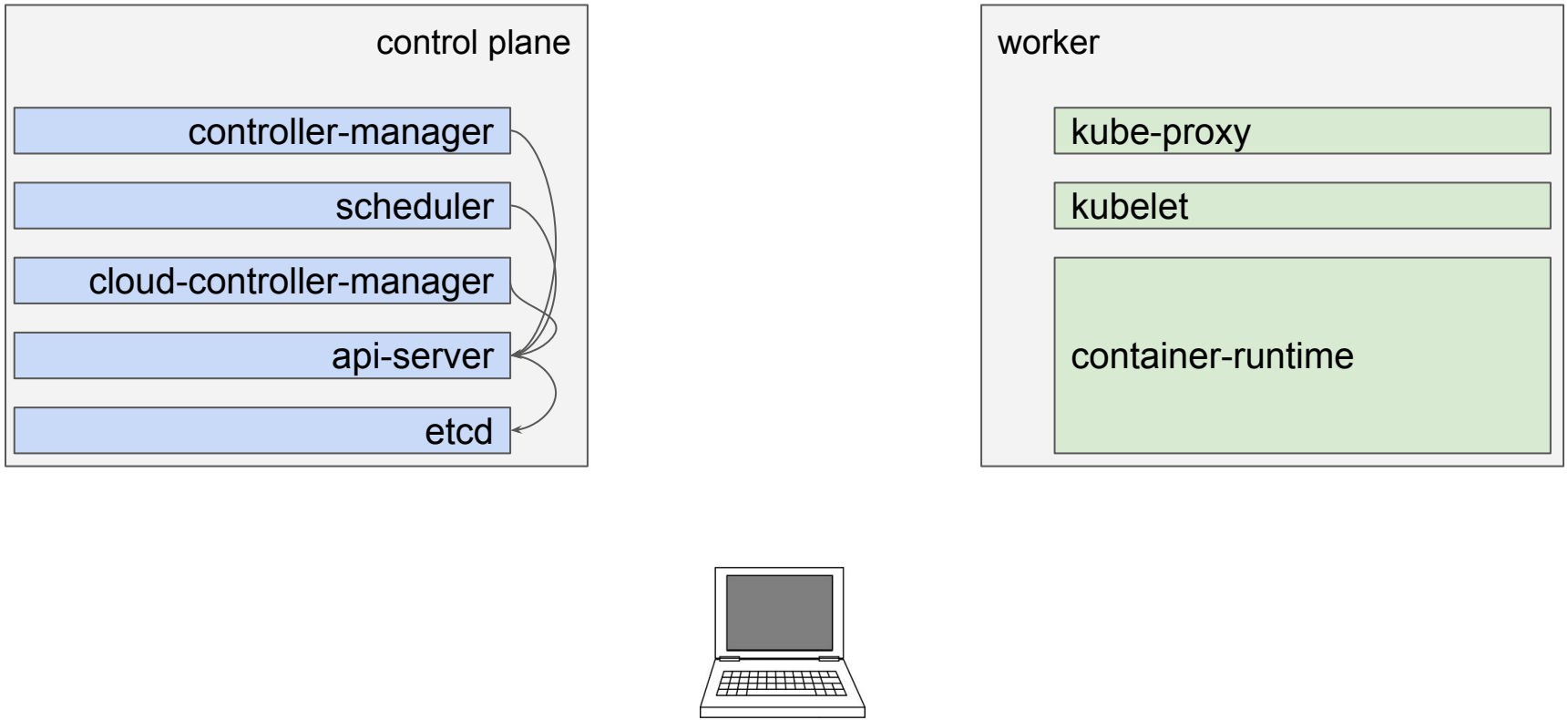


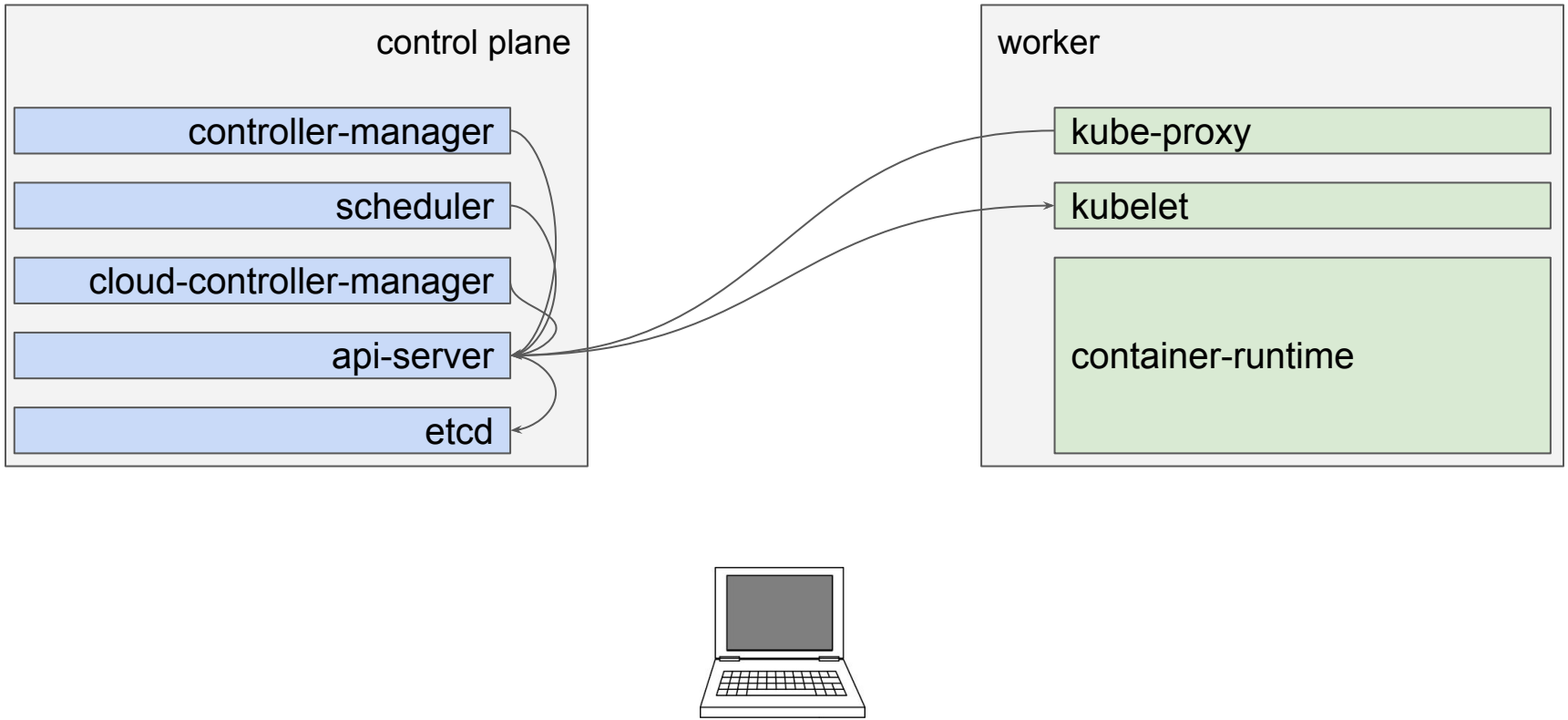

control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
container-runtime

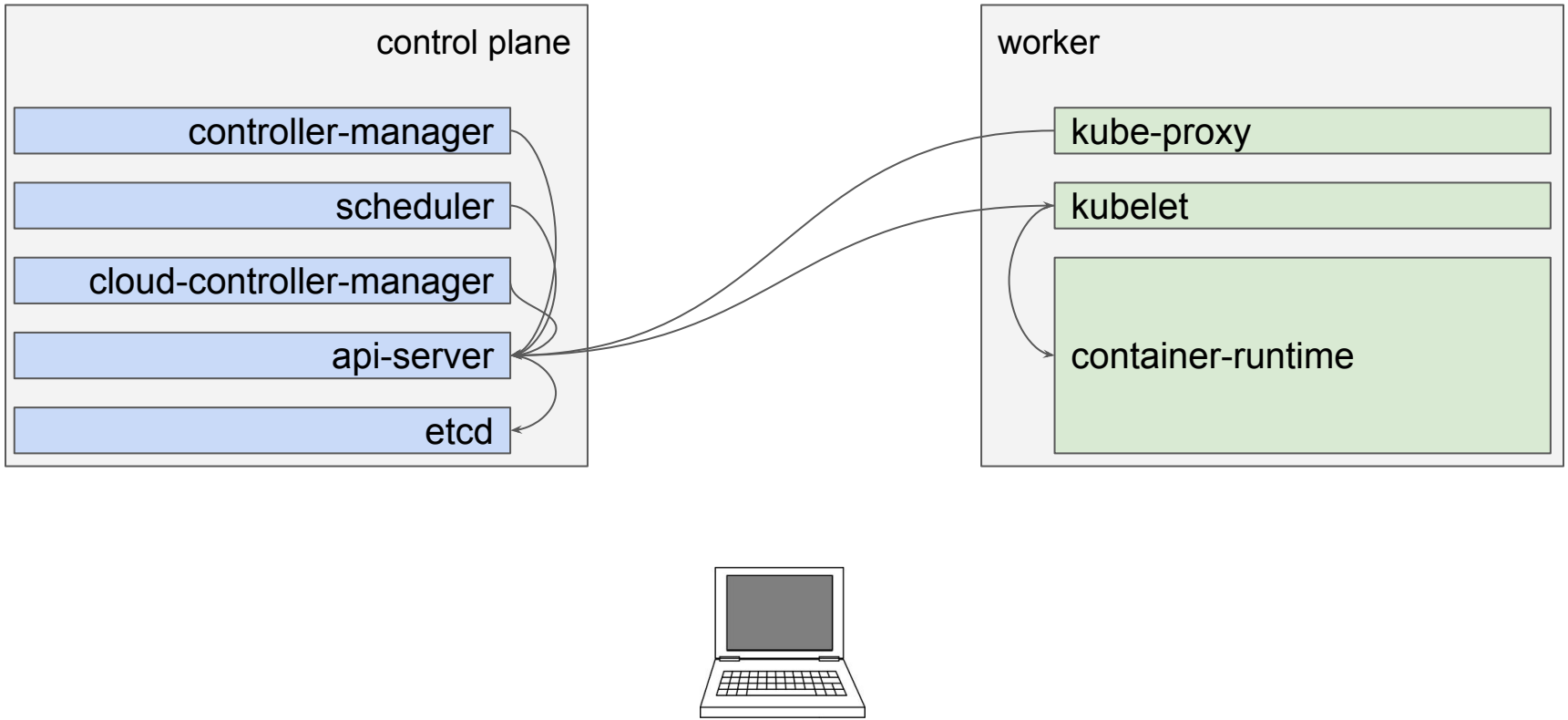

control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
kubelet
container-runtime

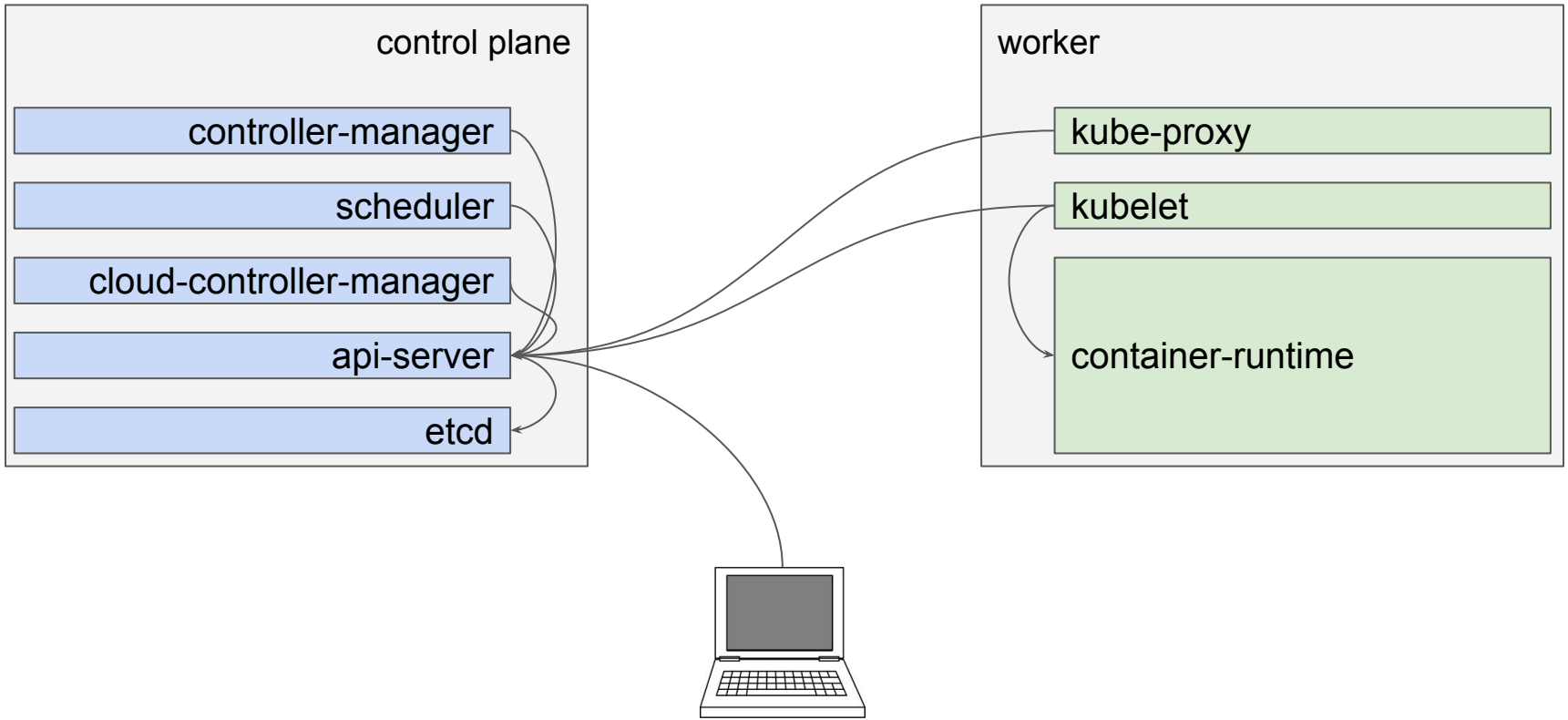

control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
kube-proxy
kubelet
container-runtime

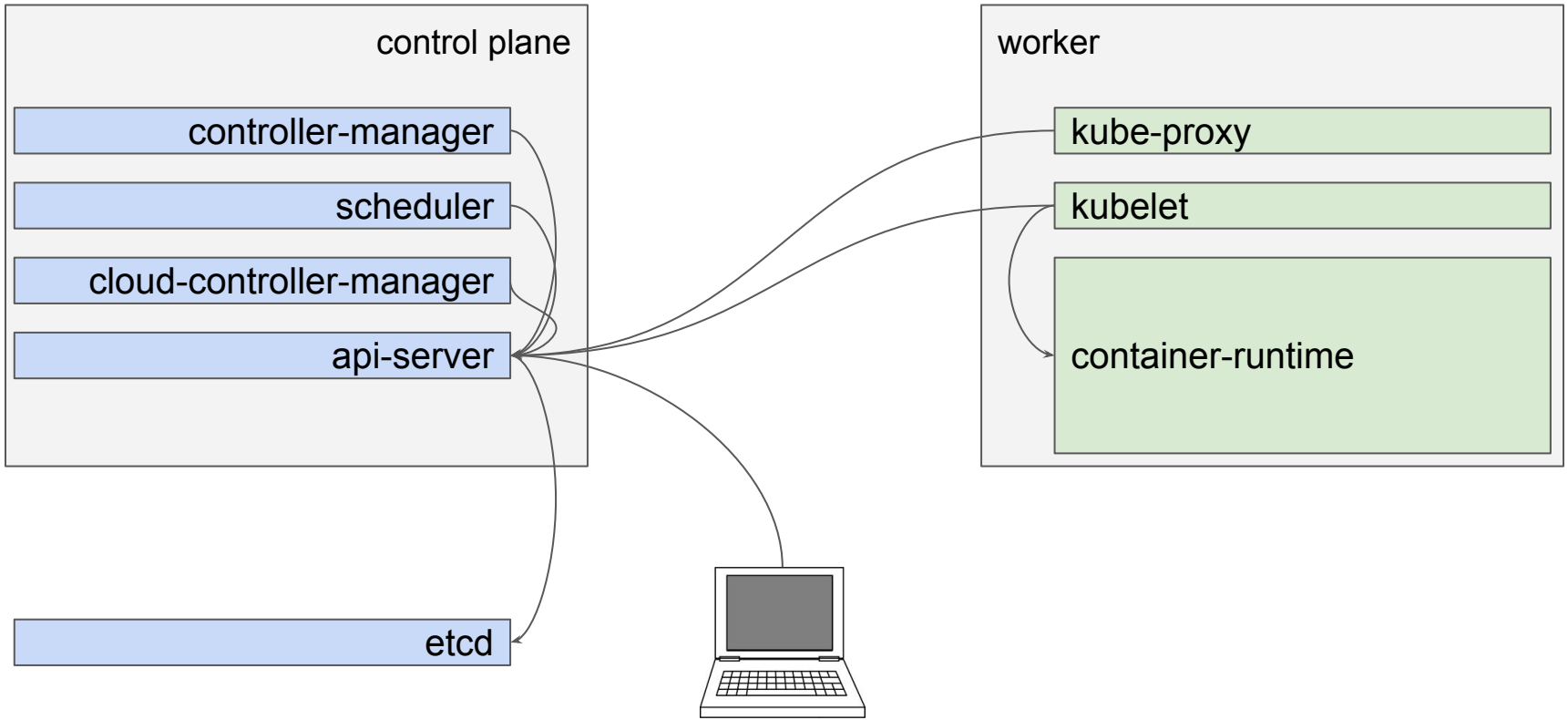

control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
kube-proxy
kubelet
container-runtime


control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
kube-proxy
kubelet
container-runtime


control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
kube-proxy
kubelet
container-runtime


control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
kube-proxy
kubelet
container-runtime

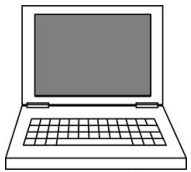

control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
kube-proxy
kubelet
container-runtime


control plane
controller-manager
scheduler
cloud-controller-manager
api-server
etcd
worker
kube-proxy
kubelet
container-runtime

control-plane-01
- controller-manager
- scheduler
- cloud-controller-manager
- api-server
- etcd

control-plane-02
- controller-manager
- scheduler
- cloud-controller-manager
- api-server
- etcd

control-plane-03
- controller-manager
- scheduler
- cloud-controller-manager
- api-server
- etcd



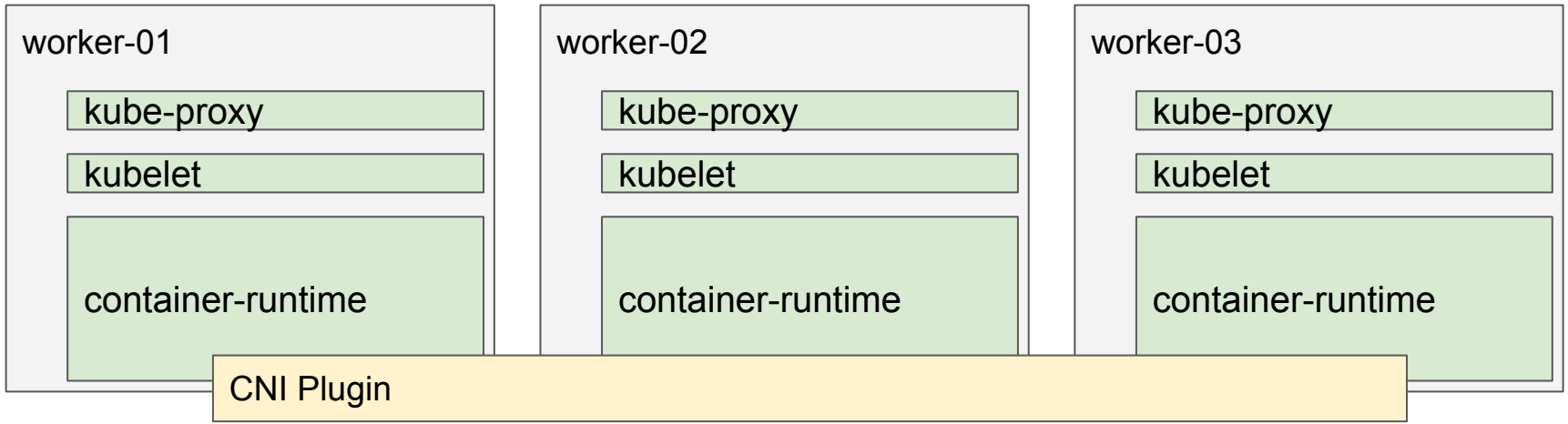
worker-01
- kube-proxy
- kubelet
- container-runtime

worker-04
- kube-proxy
- kubelet
- container-runtime

worker-02
- kube-proxy
- kubelet
- container-runtime

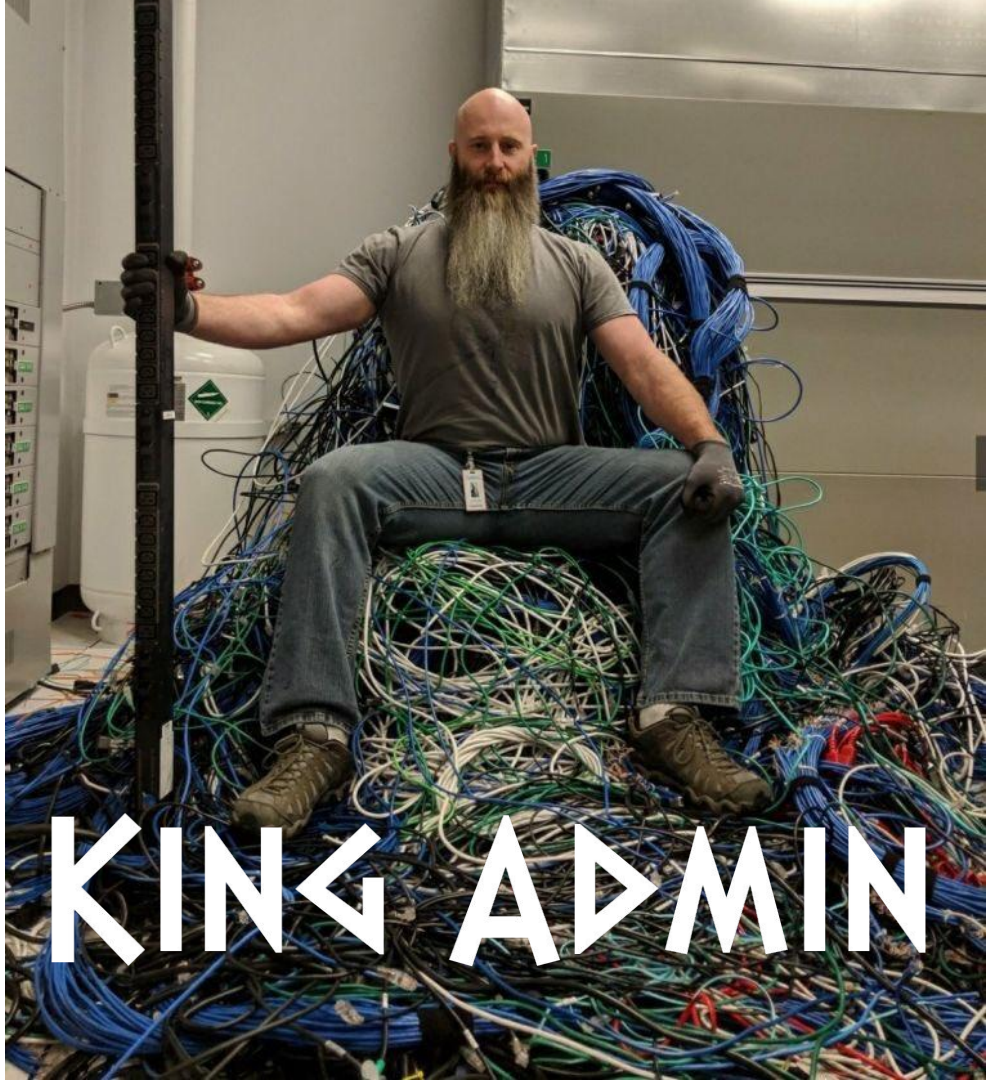
worker-05
- kube-proxy
- kubelet
- container-runtime

worker-03
- kube-proxy
- kubelet
- container-runtime

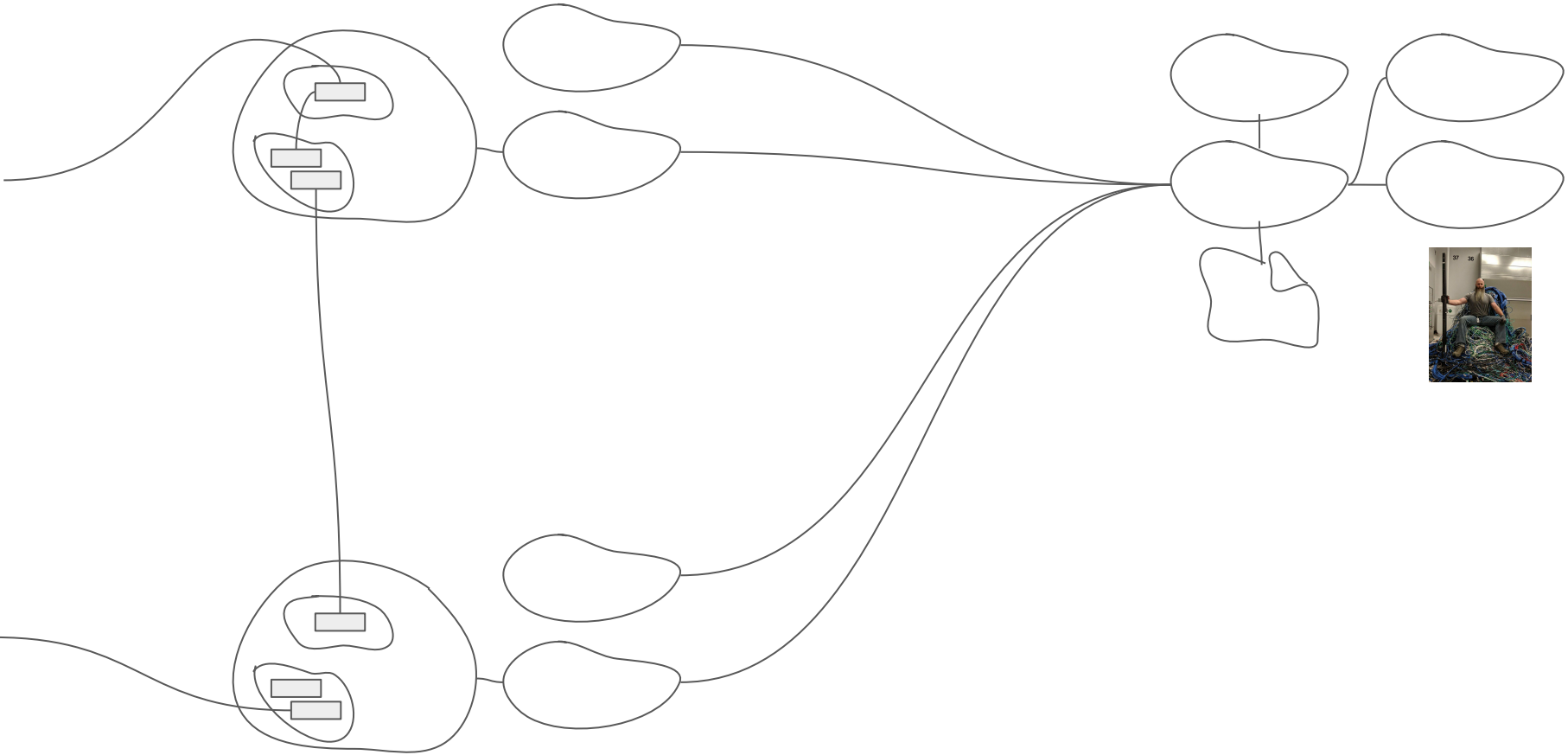

worker-01
kube-proxy
kubelet
container-runtime
worker-02
kube-proxy
kubelet
container-runtime
worker-03
kube-proxy
kubelet
container-runtime
CNI Plugin

KUBERMATIC

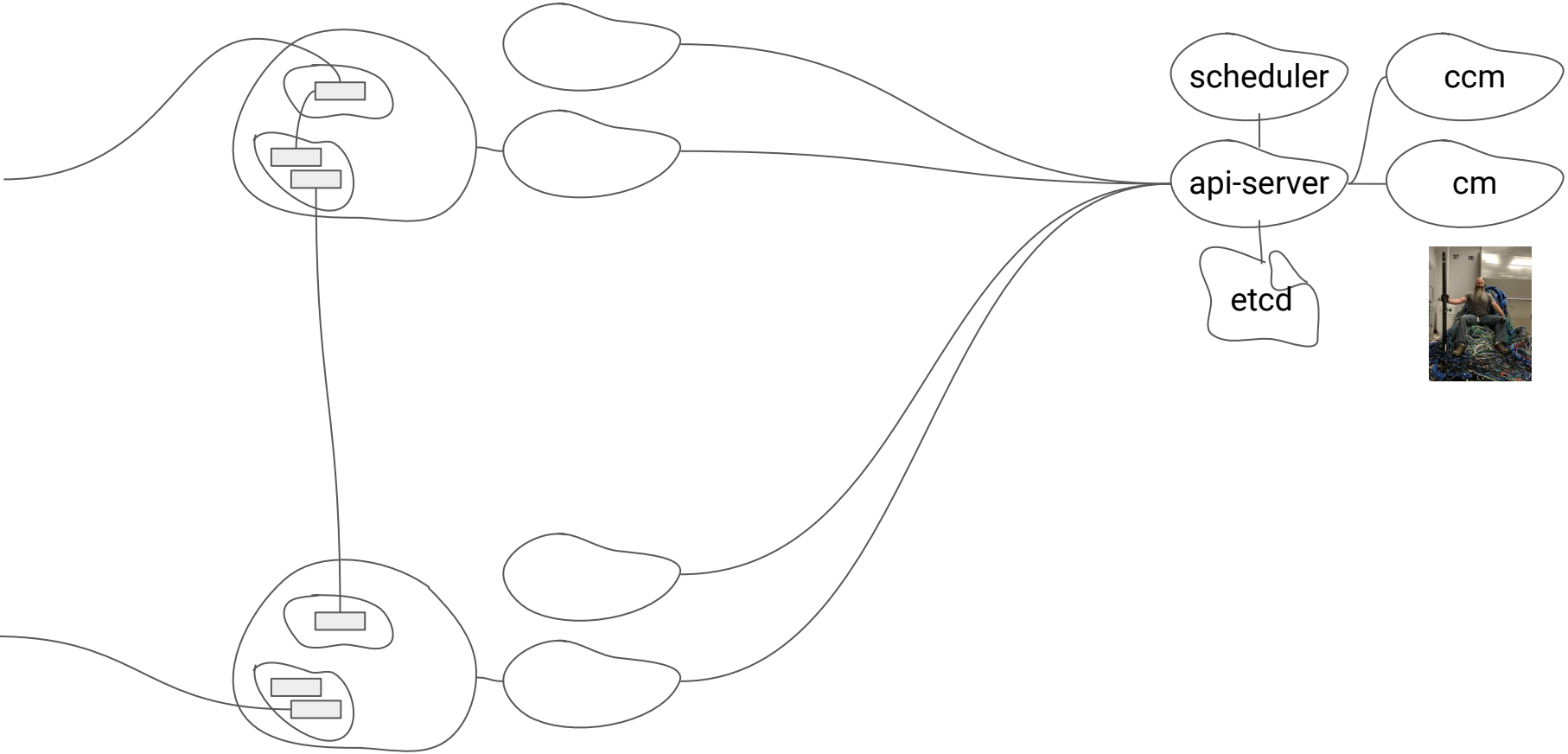
King Admin the Naive becoming King Admin the Proactive

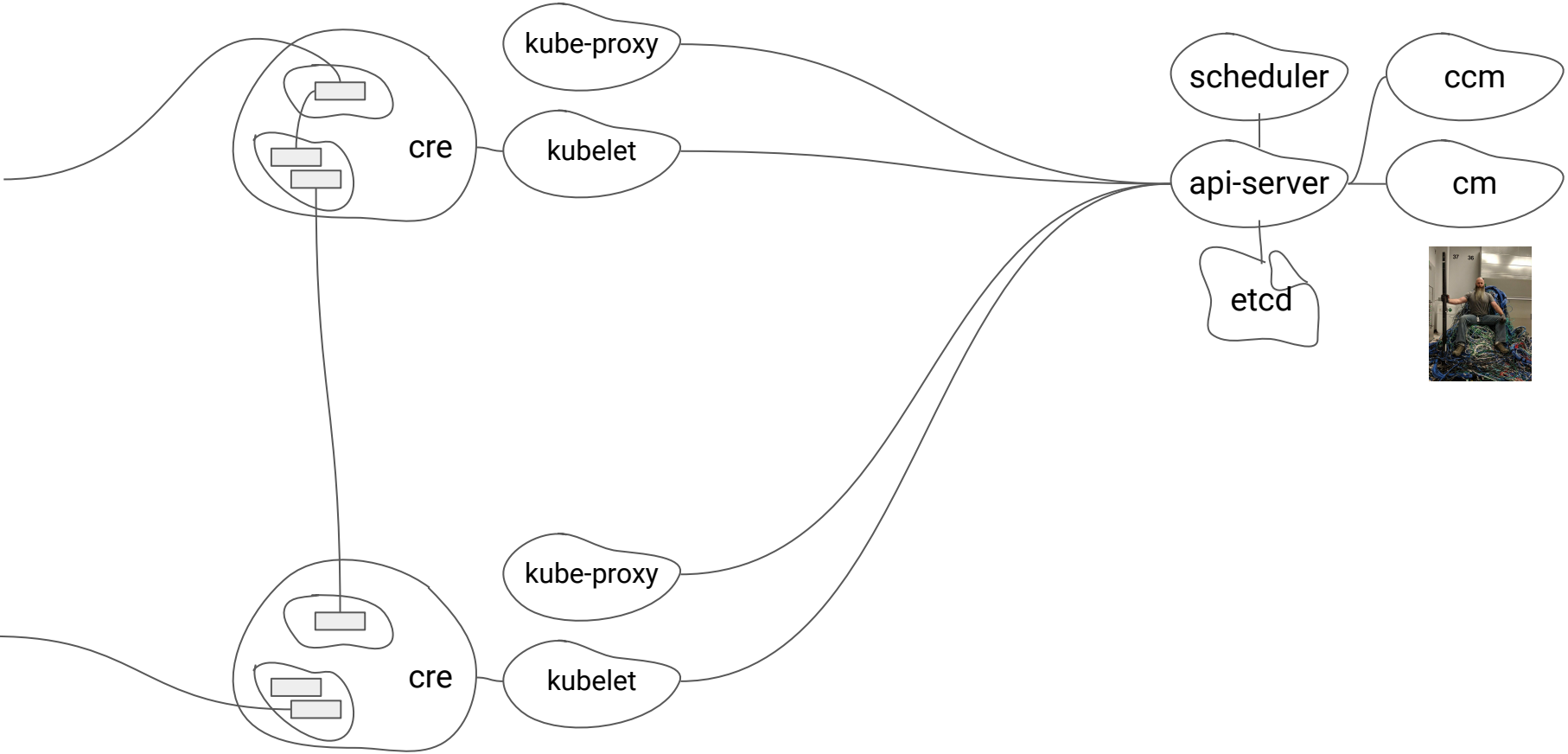

KING ADMIN

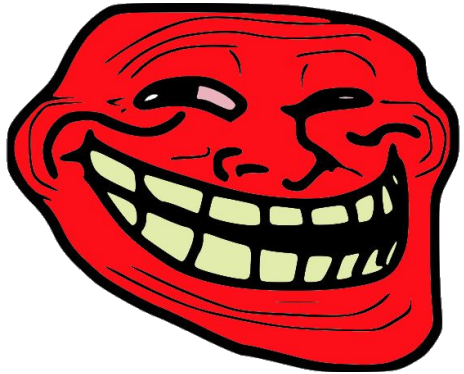
KUBOPOLIS



37
36


scheduler
ccm
api-server
cm
etcd


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

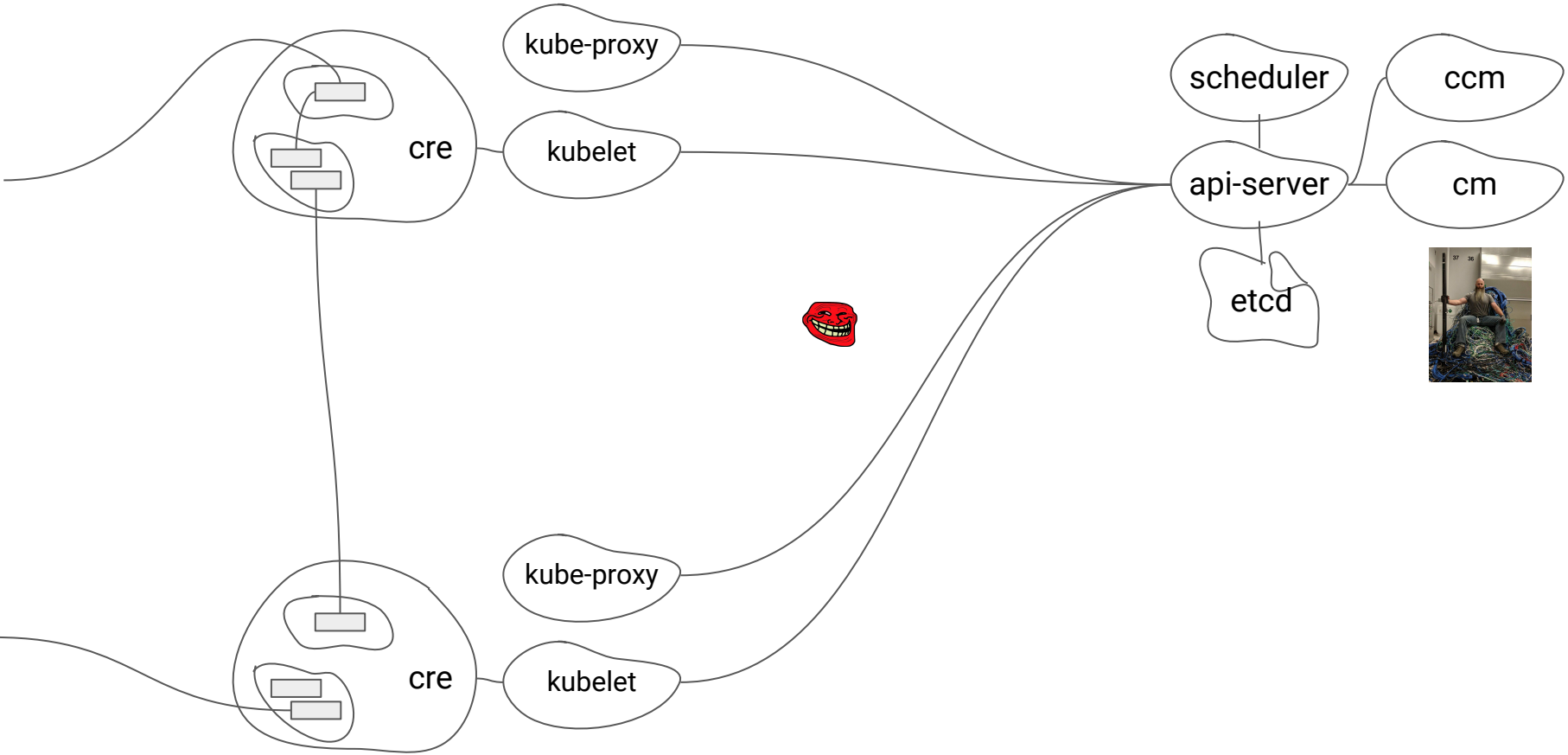


HAKKERIOS

KUBERMATIC

Hakkerios attacks everywhere

Firewalls

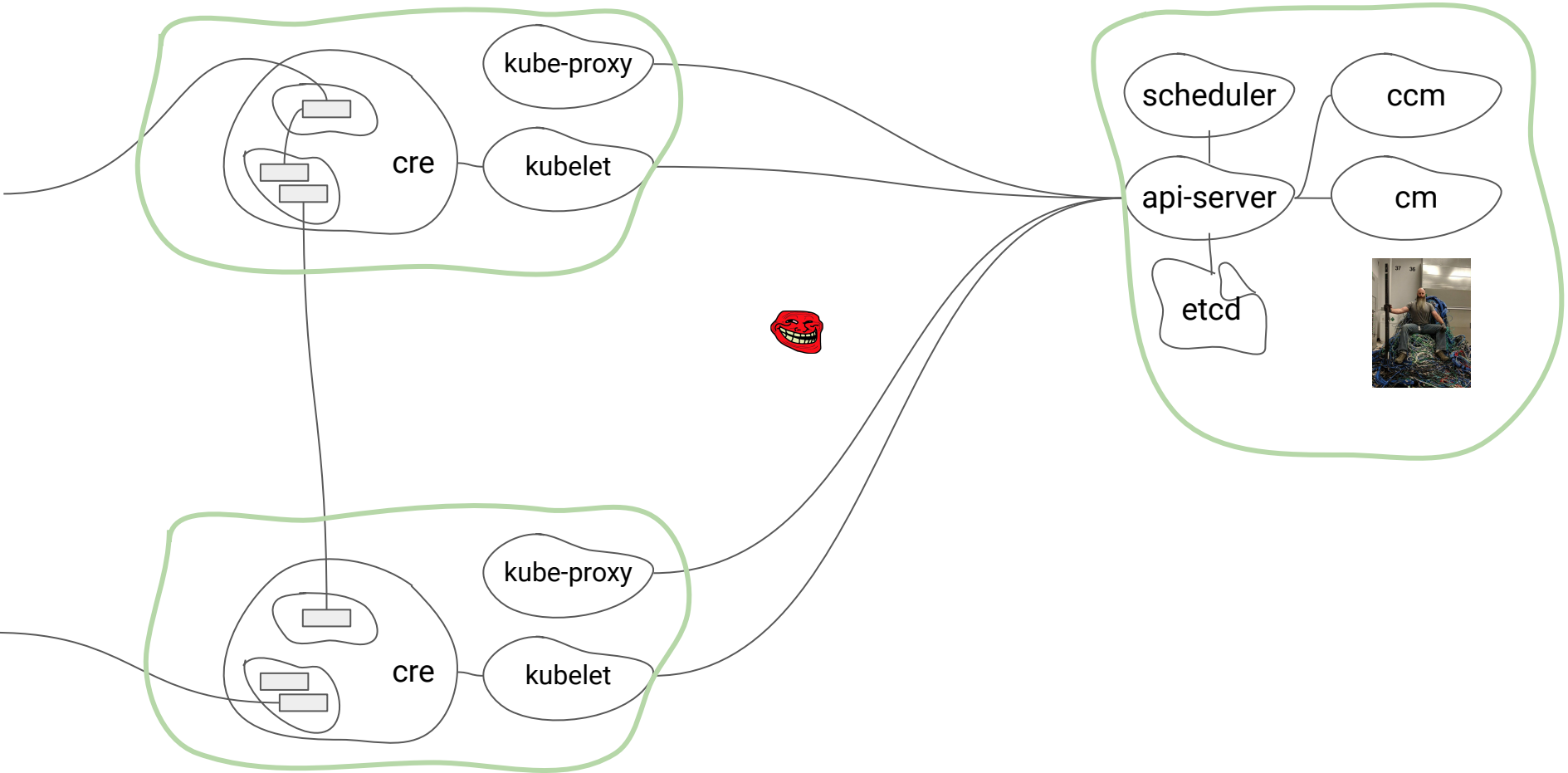

kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

Kubernetes Control Plane Ports

- **`6443`**: api-server
- **`2379-2380`**: etcd
- **`10250`**: kubelet
- **`10259`**: kube-scheduler
- **`10257`**: kube-controller-manager

Kubernetes Worker Nodes Ports

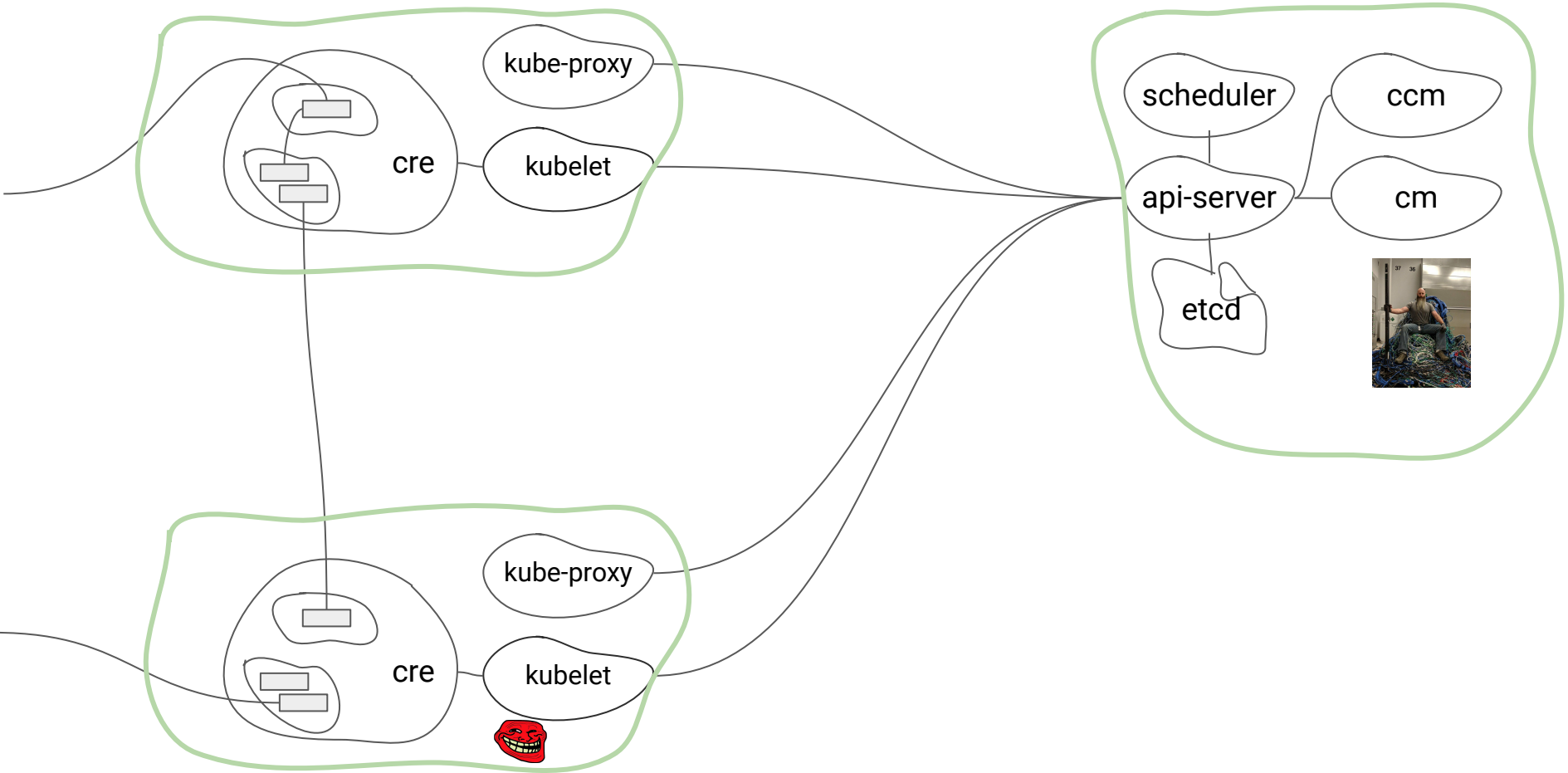
- `10250`: kubelet
- `30000-32767`: default port range for NodePort Services


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

KUBERMATIC

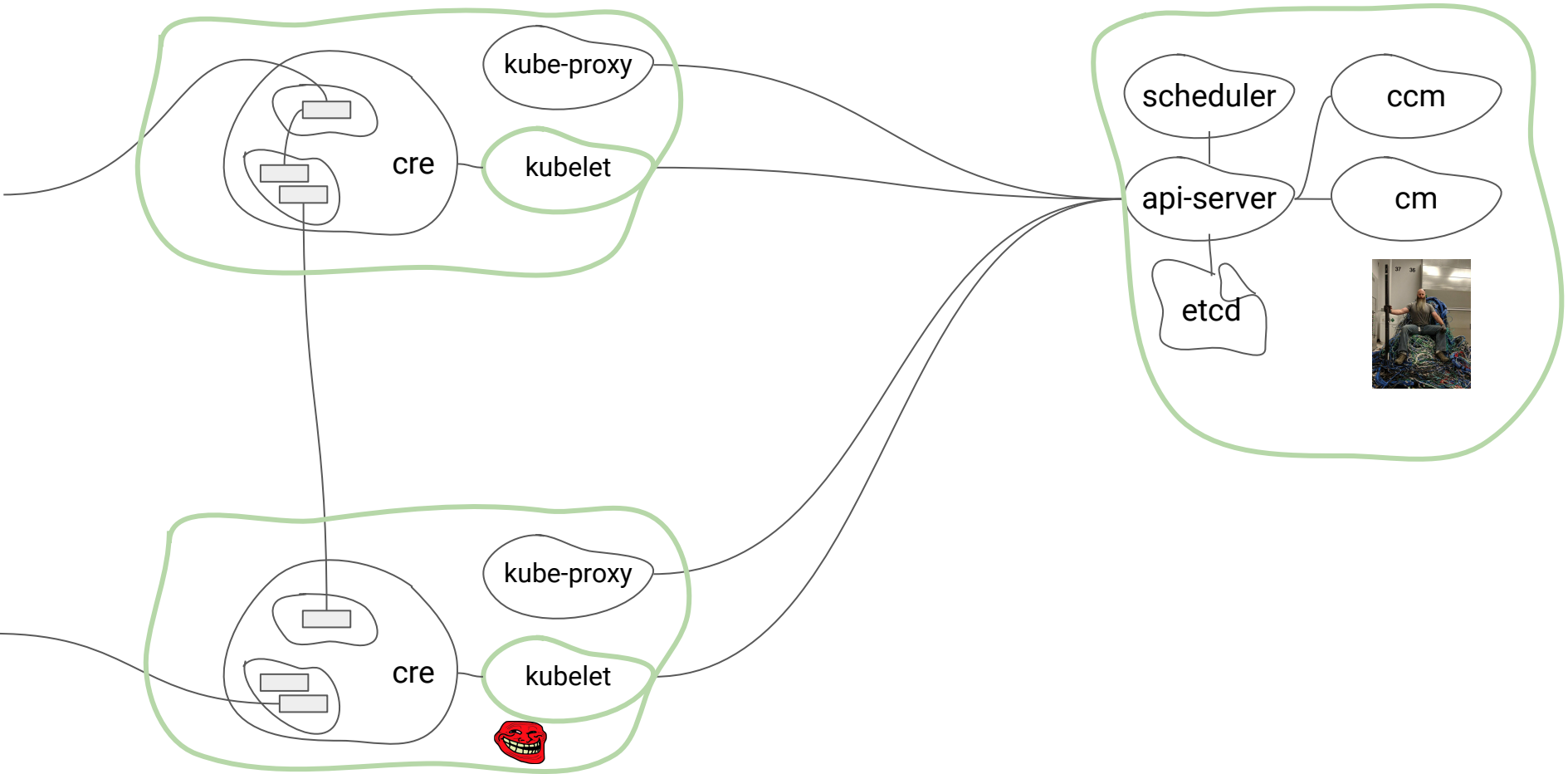
Hakkerios attacks the suburbs

Securing the worker nodes


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

Kubelet

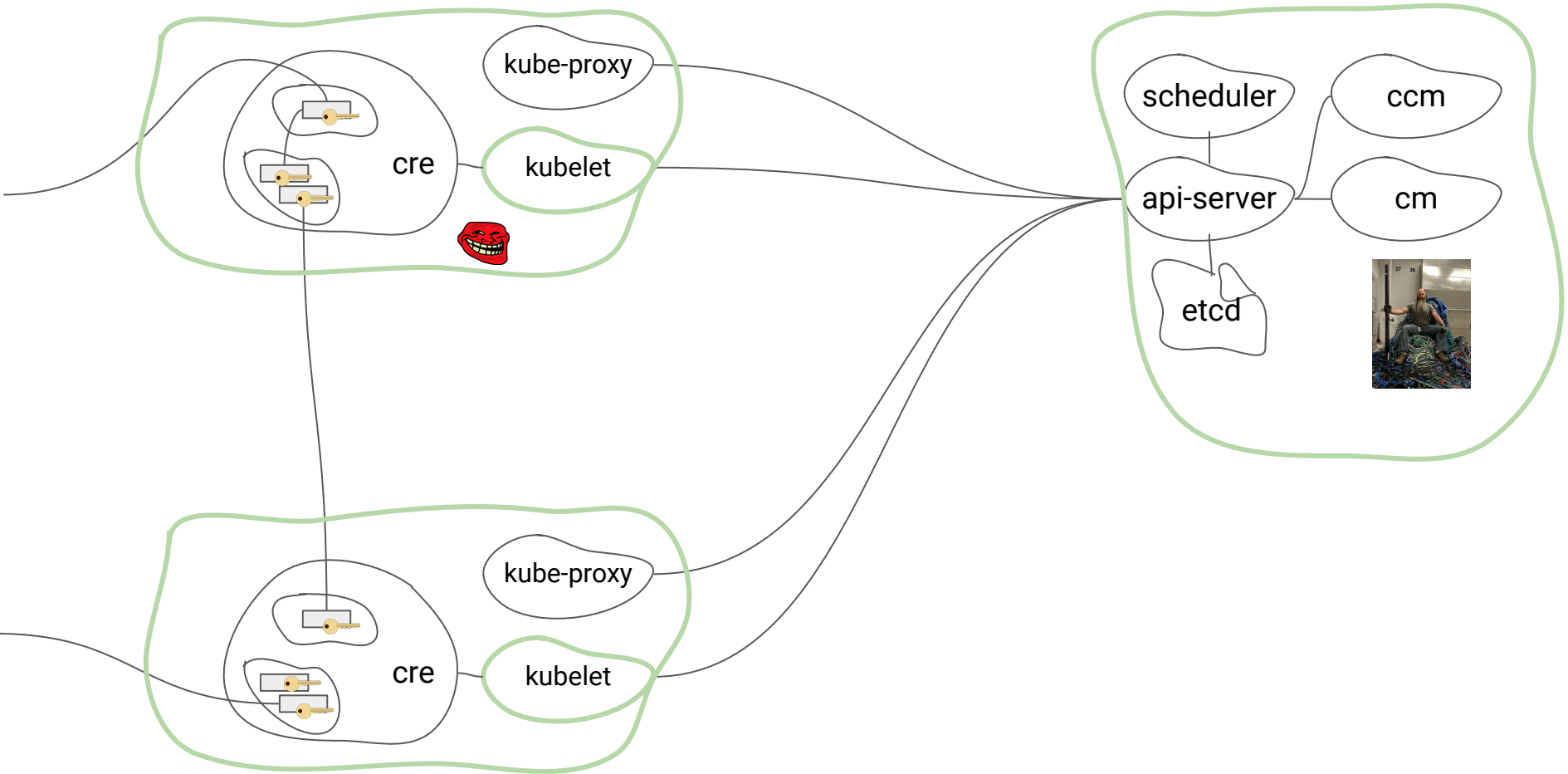
- <= Getting the PodSpecs From the API-Server
- => Fetching logs for pods
- => Attaching (usually through kubectl) to running pods
- => Providing the kubelet's port-forwarding functionality


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

KUBERMATIC

Hakkerios identity theft

Restricting permissions to users / applications


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

Is a Pod having a ServiceAccount if you haven't defined one in your PodSpec?

Avoid storing Service Account Tokens in Containers

```
...                                  # pod.yaml
spec:
  terminationGracePeriodSeconds: 0
  automountServiceAccountToken: false
  containers:
    - name: my-ubuntu
      image: ubuntu
...
```

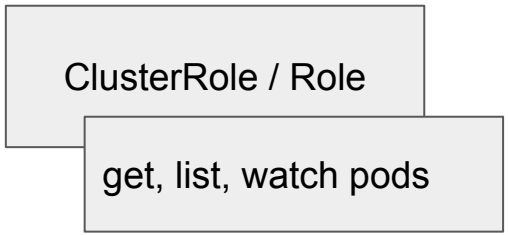
Kubernetes Authorization Modes

- **Node**
  - Node authorization is a special-purpose authorization mode that specifically authorizes API requests made by kubelets.
- **ABAC**
  - Attribute-based access control
  - Json file which has to be taught towards the api-server
  - Rarely used nowadays
- **RBAC**
  - Role, ClusterRole, RoleBinding and ClusterRoleBinding
  - The default nowadays
- **Webhook**

You can combine authorization modes

```
...
    - kube-apiserver
        - --advertise-address=195.201.225.34
        - --allow-privileged=true
        - --authorization-mode=Node,RBAC
        - --client-ca-file=/etc/kubernetes/pki/ca.crt
        - --enable-admission-plugins=NodeRestriction
        - --enable-bootstrap-token-auth=true
...
```

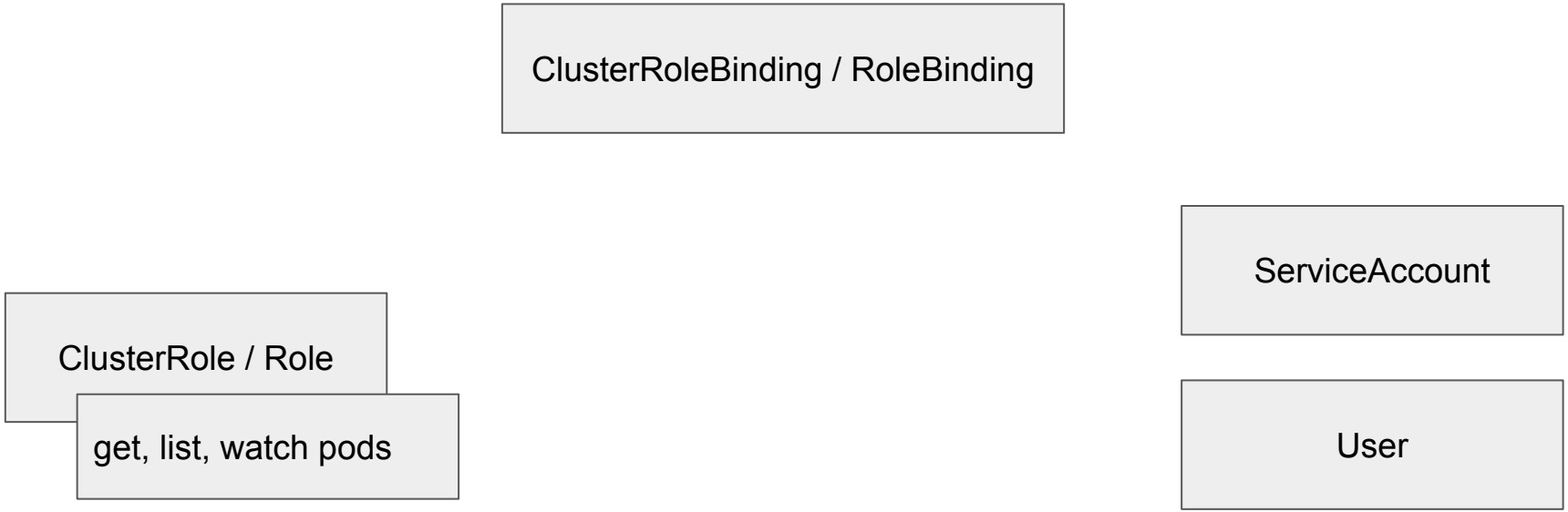
RBAC



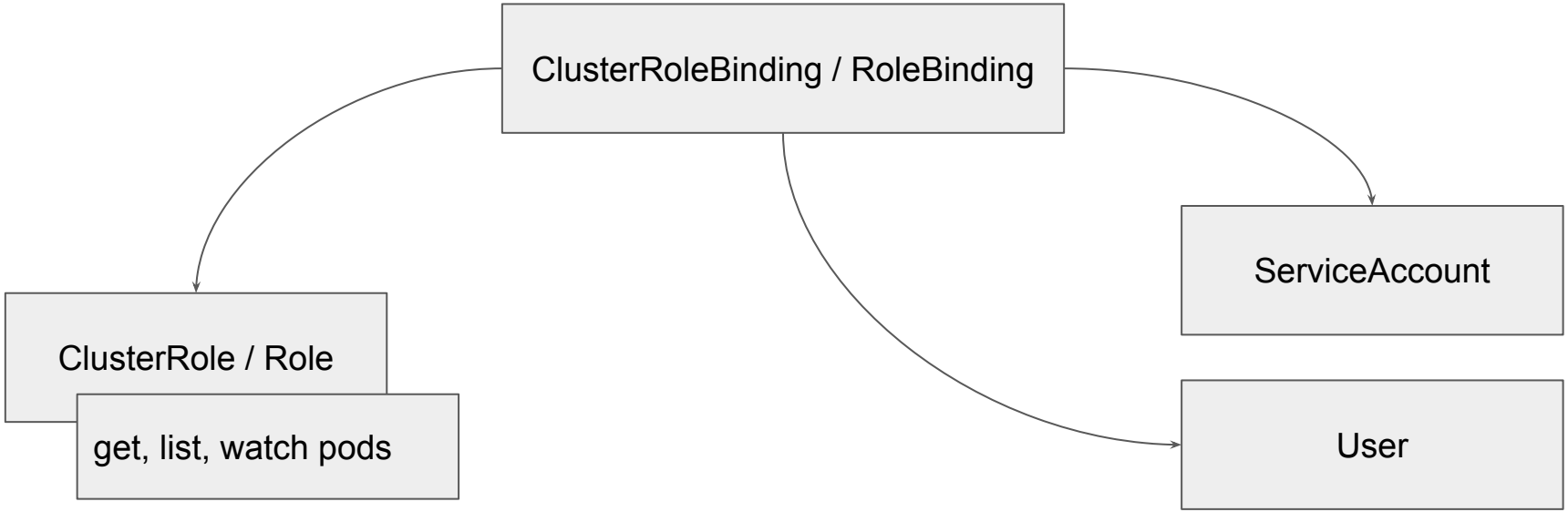
RBAC

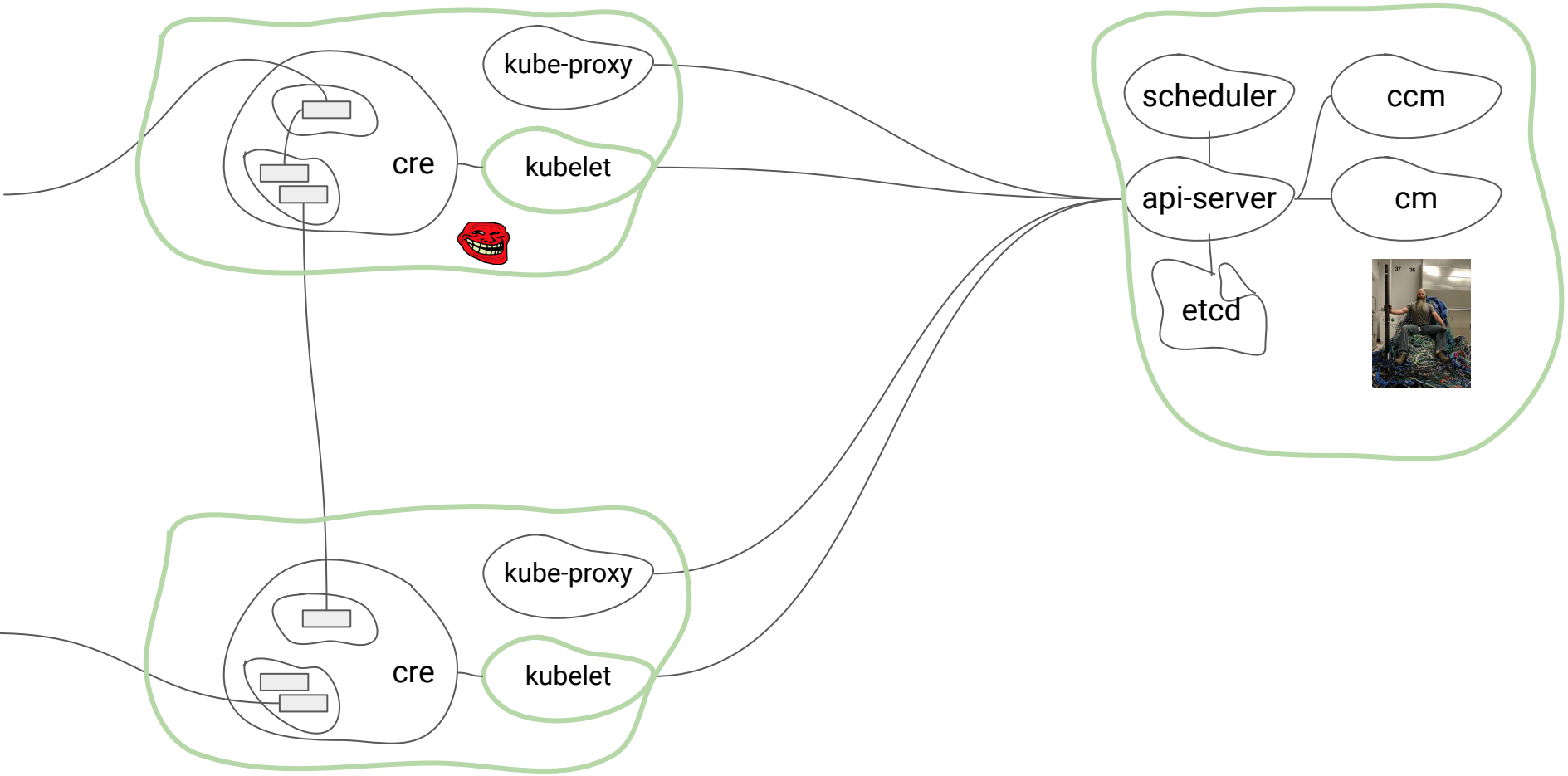
RBAC

RBAC


ClusterRoleBinding / RoleBinding
ServiceAccount
ClusterRole / Role
get, list, watch pods
User

RBAC

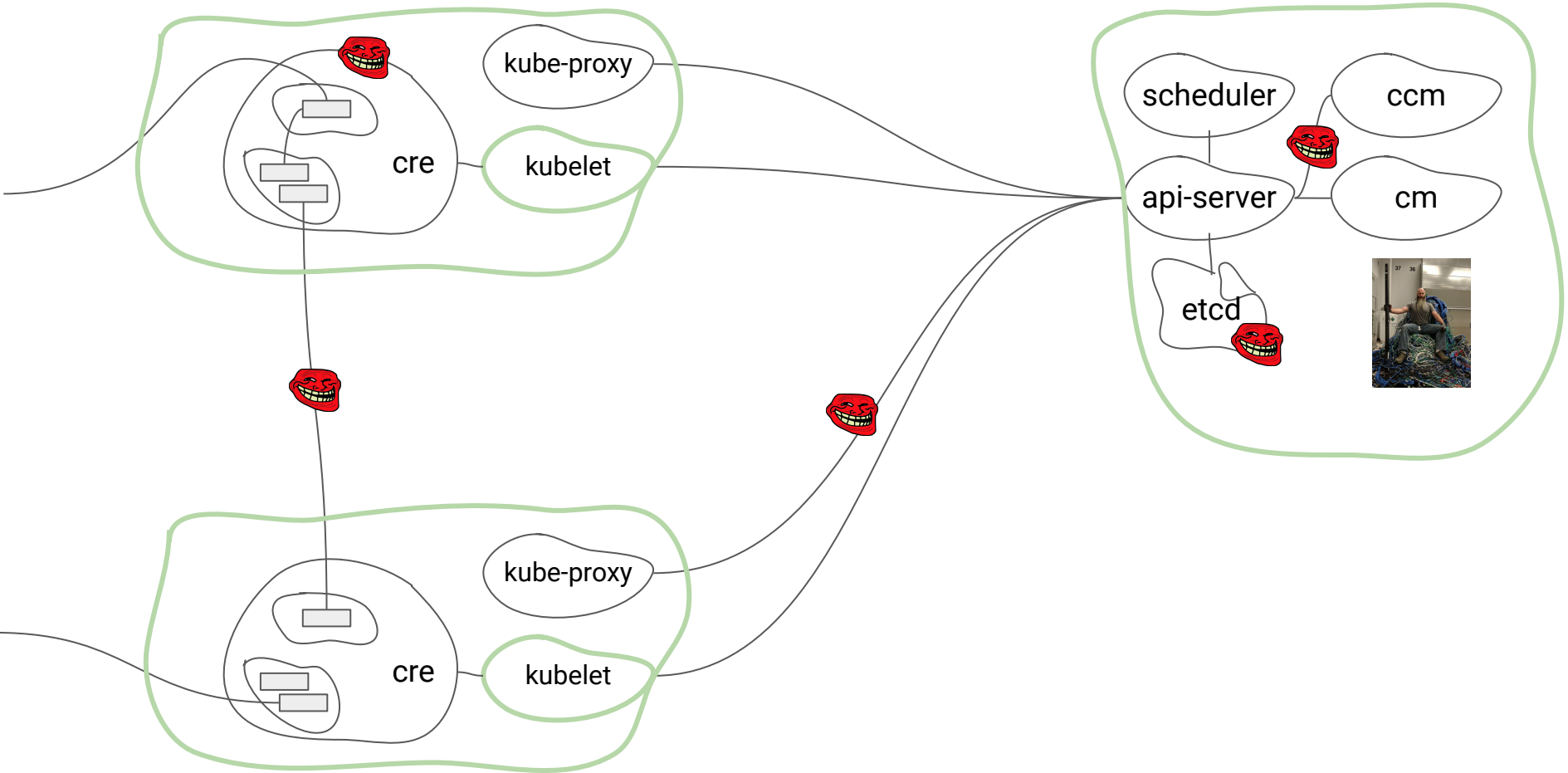

ClusterRoleBinding / RoleBinding
ClusterRole / Role
get, list, watch pods
ServiceAccount
User


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

KUBERMATIC

Hakkerios attacks messengers

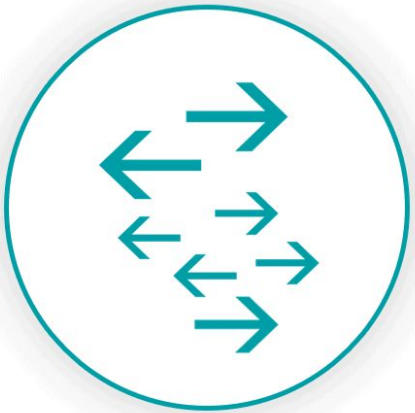
Encryption at Usage / Transport / Rest


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

THE THREE STATES OF DATA

AT REST

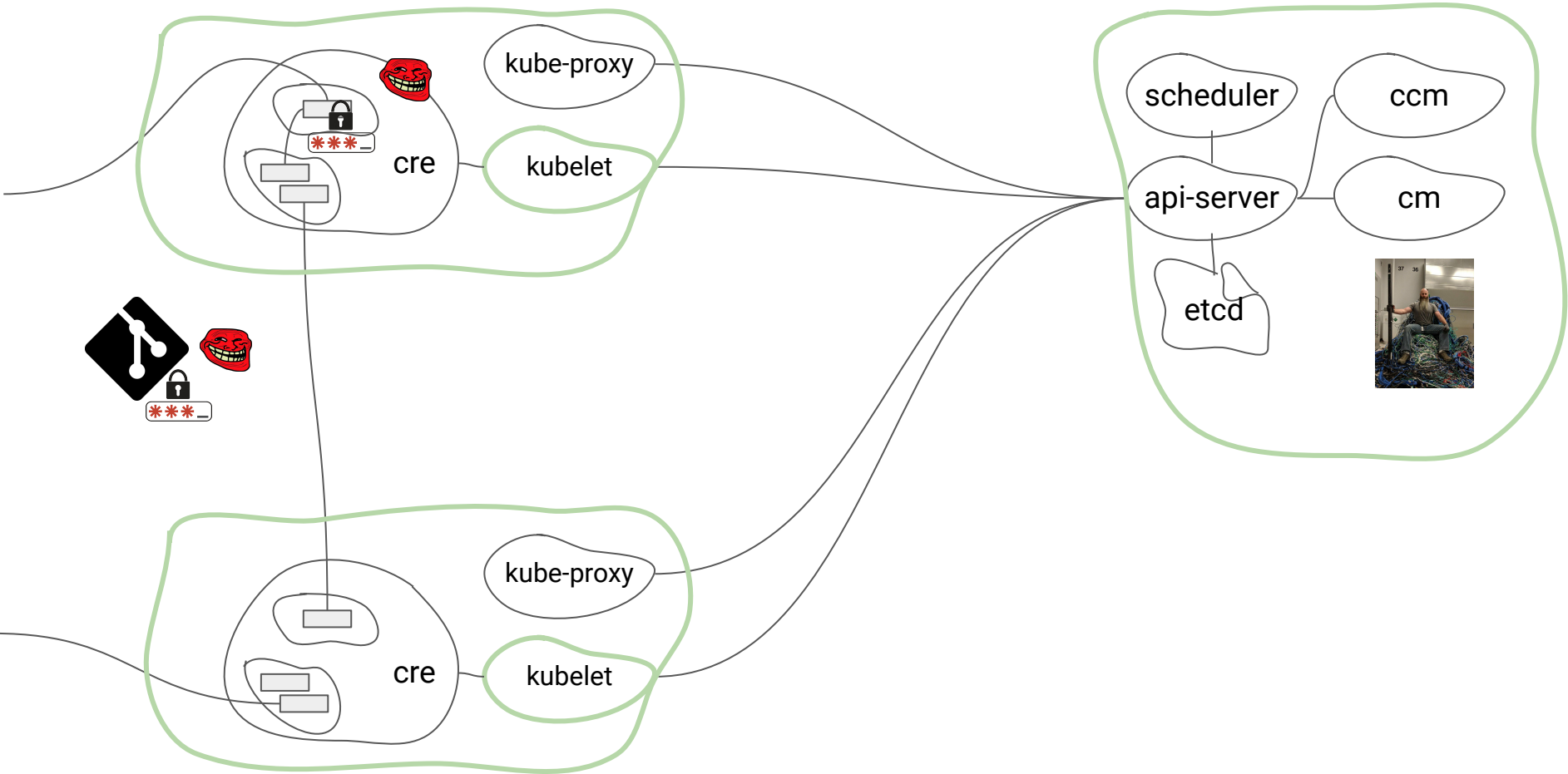
IN TRANSIT



IN USE

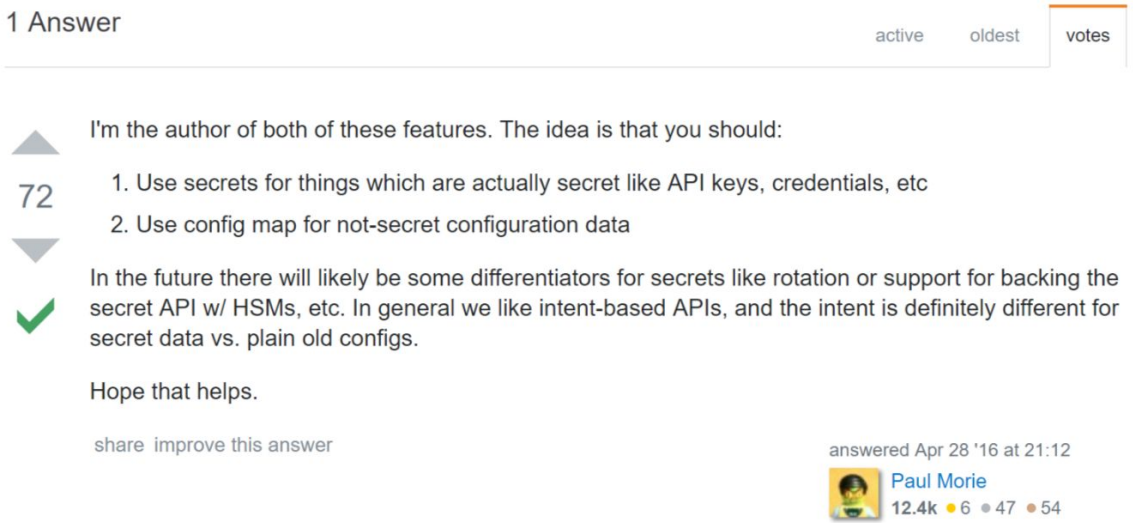
KUBERMATIC

Encryption at Usage


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

Secrets

- Secrets are NOT encrypted, they are base64 encoded
- A tmpfs gets mounted into the pods, they do not touch a HDD
- Within the Pods the Secrets are base64 decoded!



1 Answer

active oldest votes

72

I'm the author of both of these features. The idea is that you should:

1. Use secrets for things which are actually secret like API keys, credentials, etc
2. Use config map for not-secret configuration data

In the future there will likely be some differentiators for secrets like rotation or support for backing the secret API w/ HSMs, etc. In general we like intent-based APIs, and the intent is definitely different for secret data vs. plain old configs.

Hope that helps.

share improve this answer

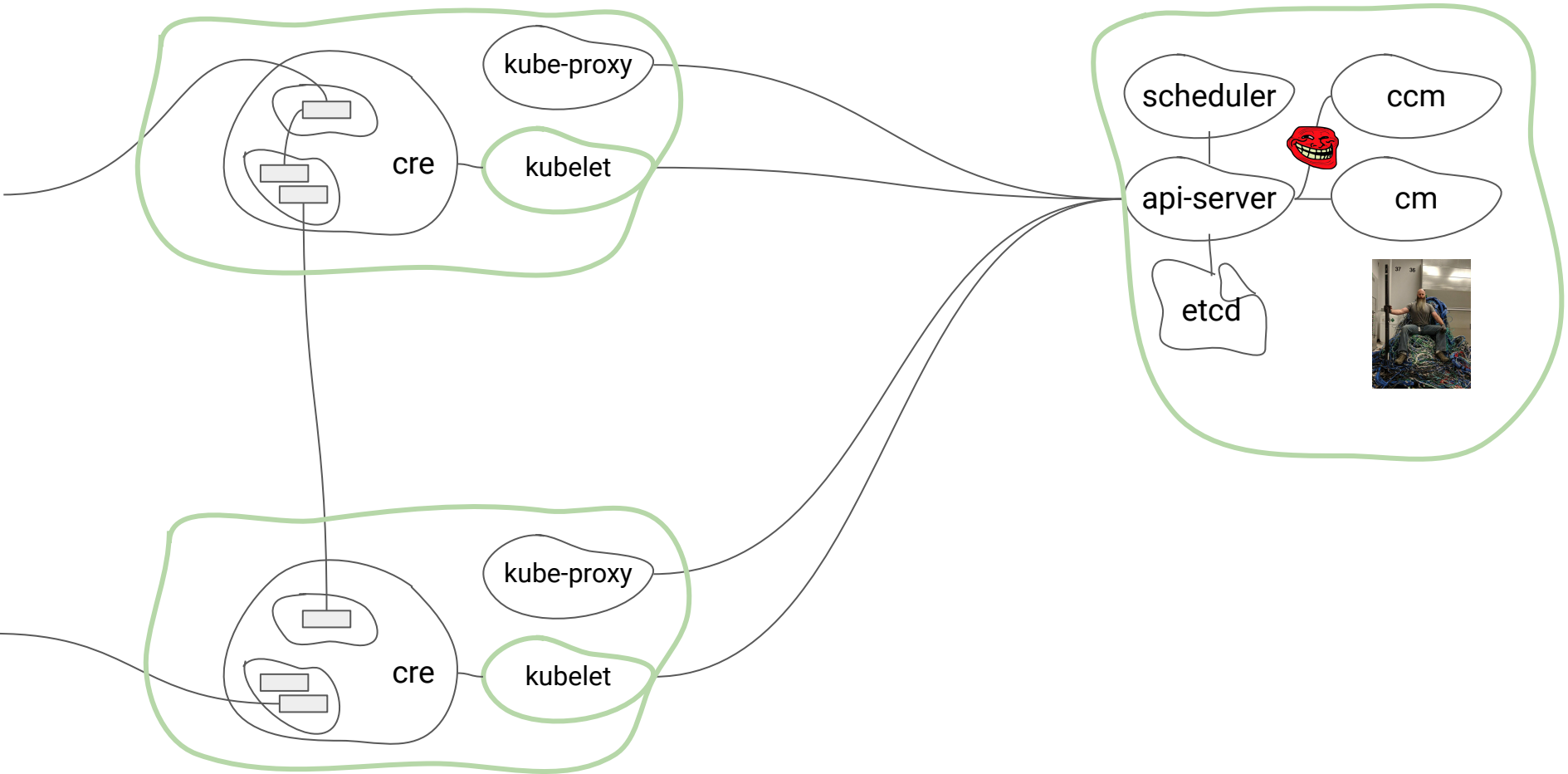
answered Apr 28 '16 at 21:12

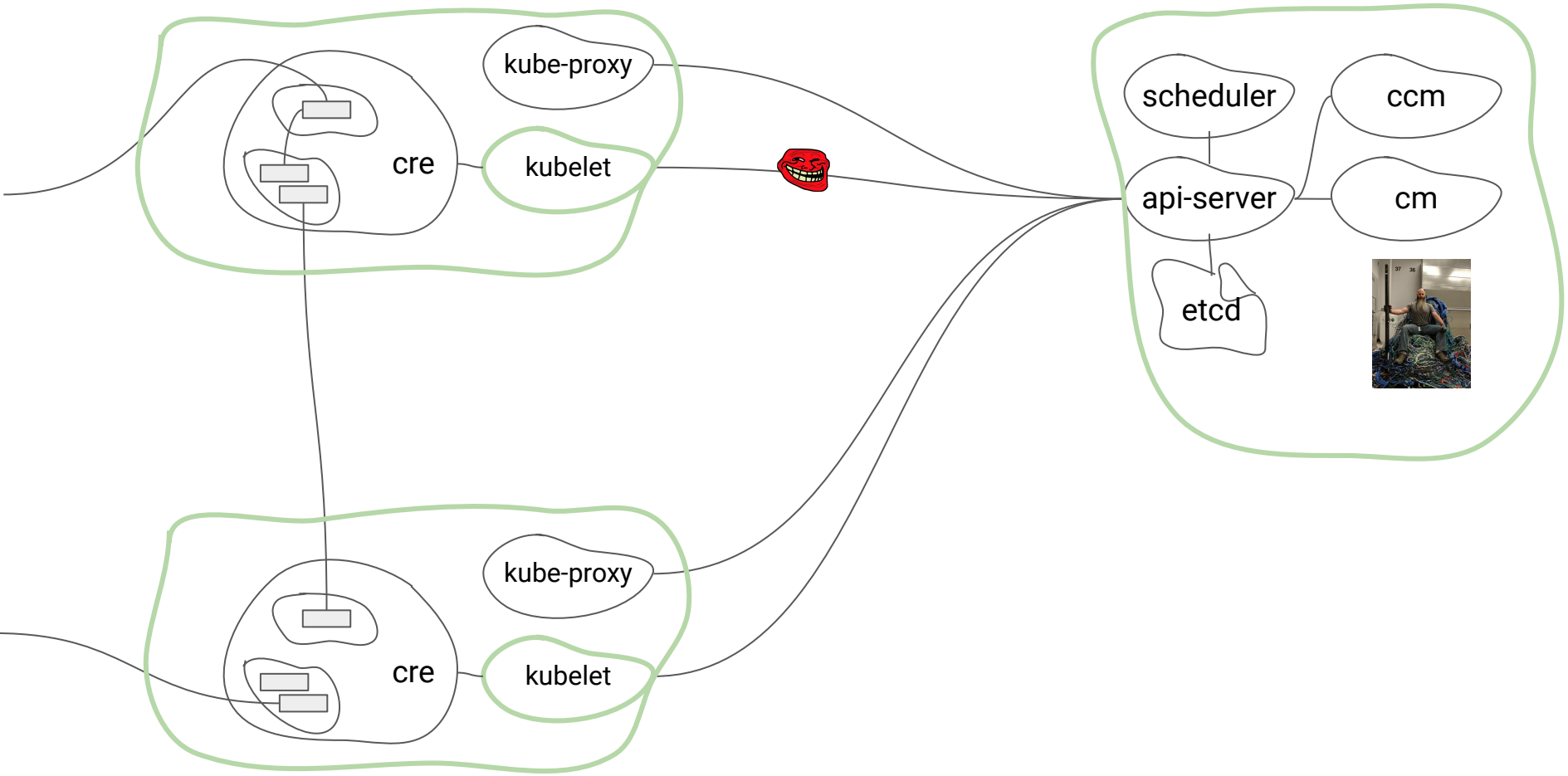
Paul Morie

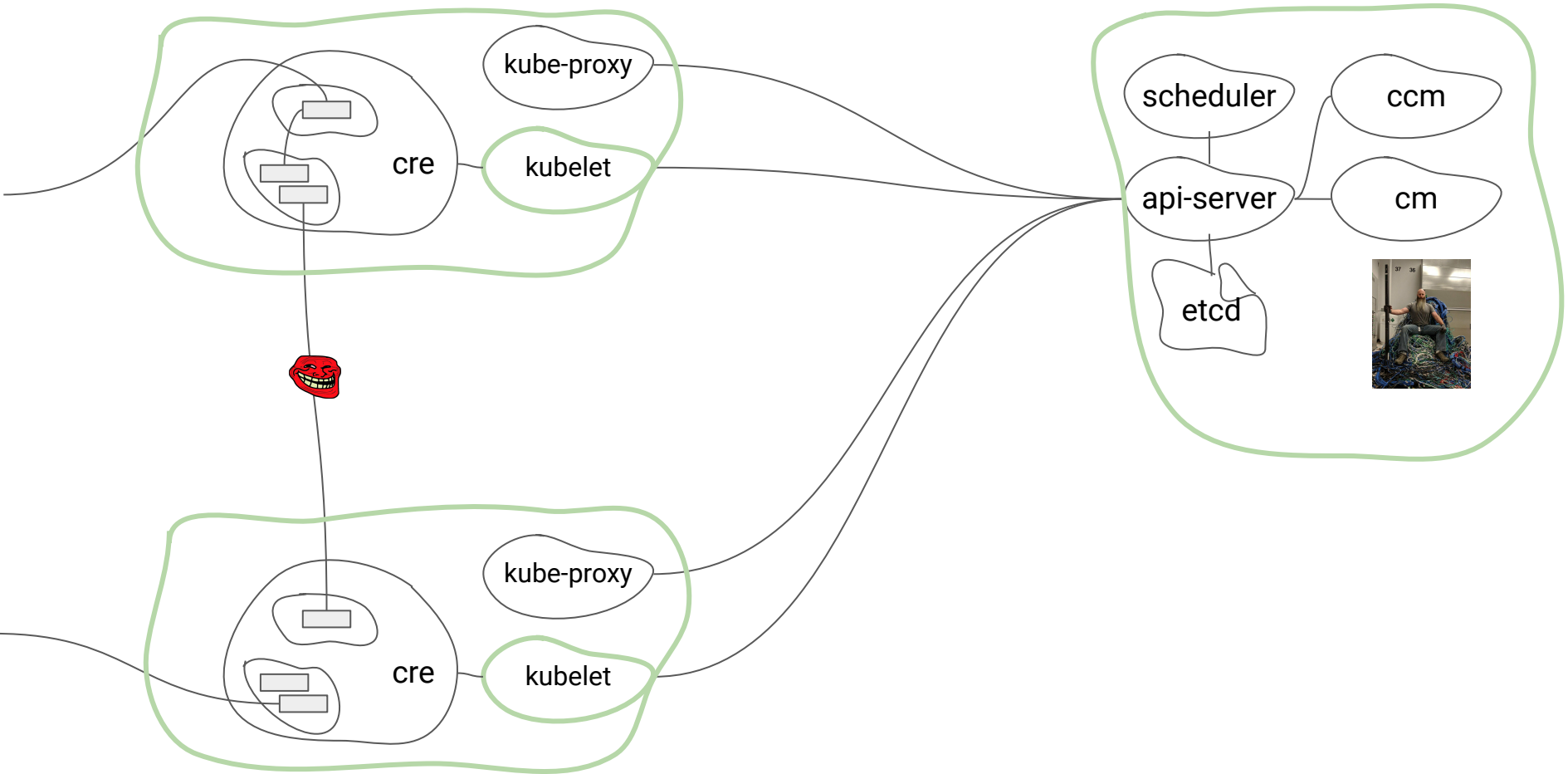
12.4k ● 6 ● 47 ● 54

KUBERMATIC

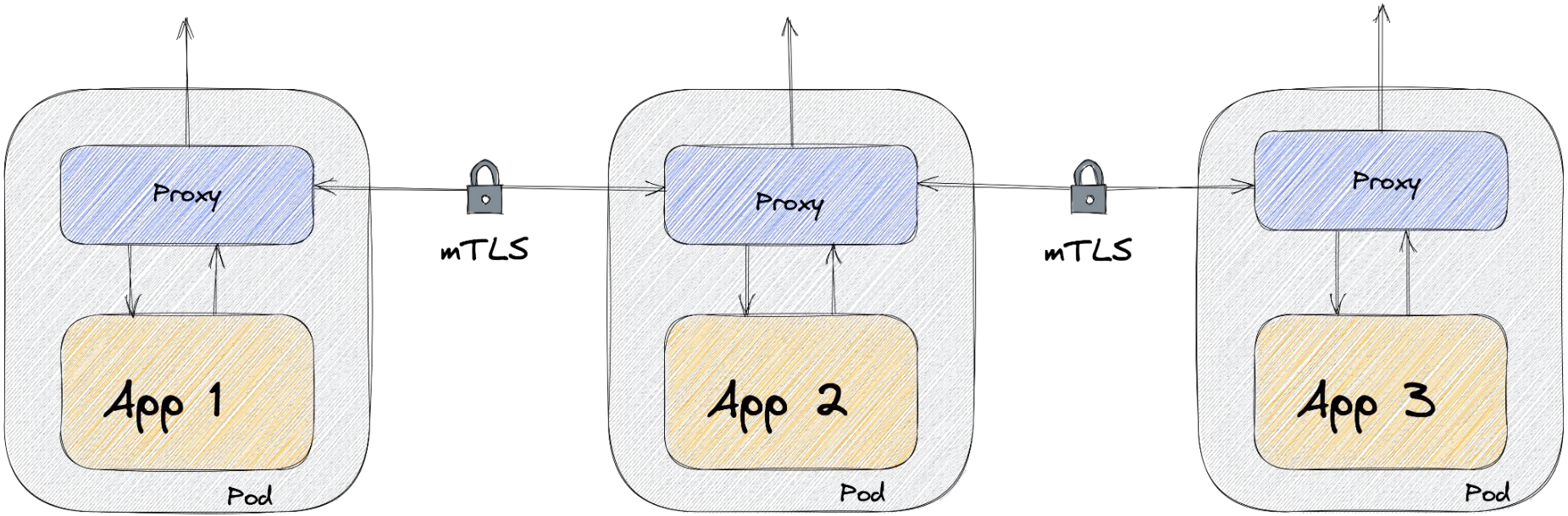
Encryption at Transport


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

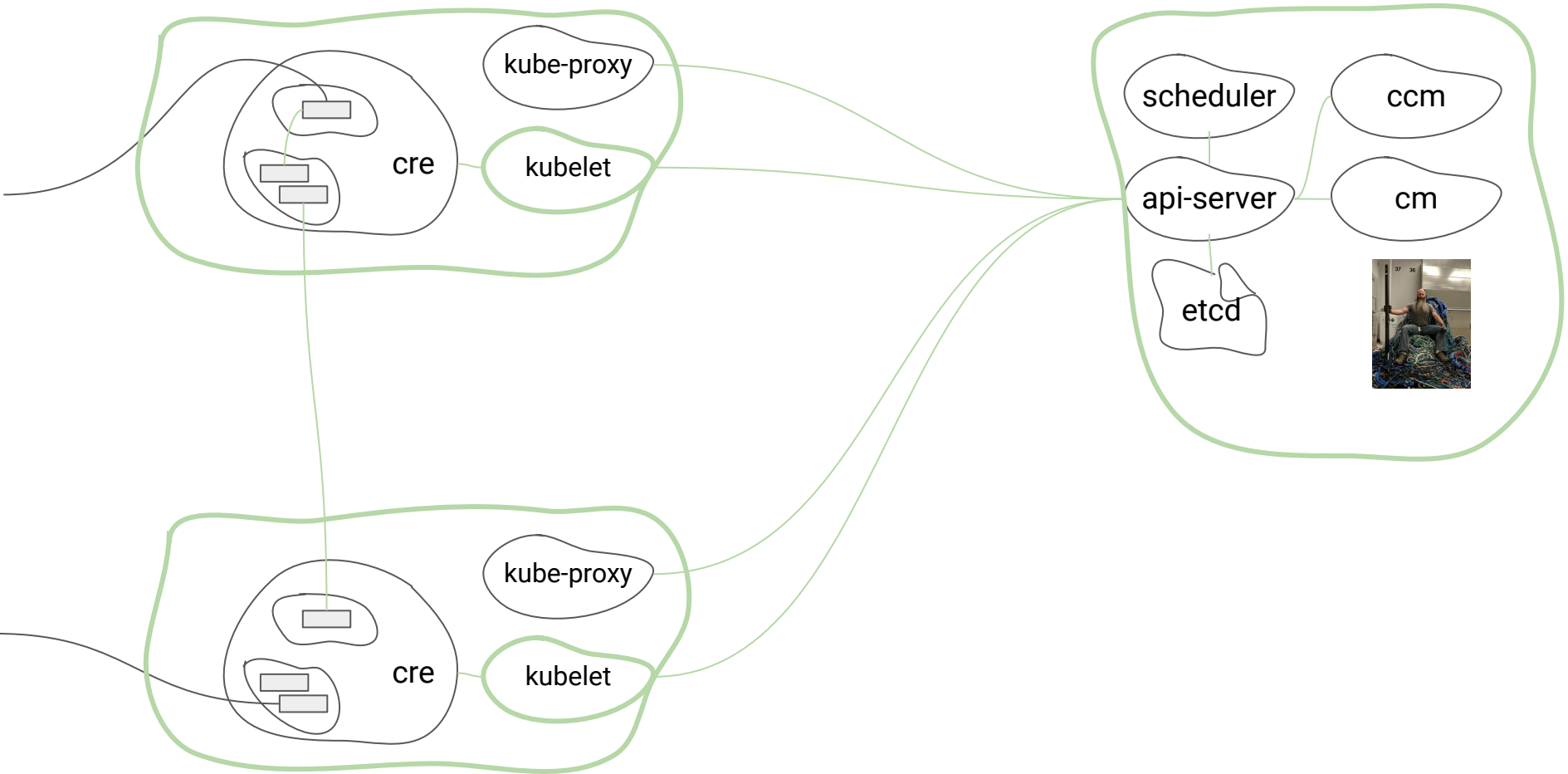

kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

mTLS via Service Mesh



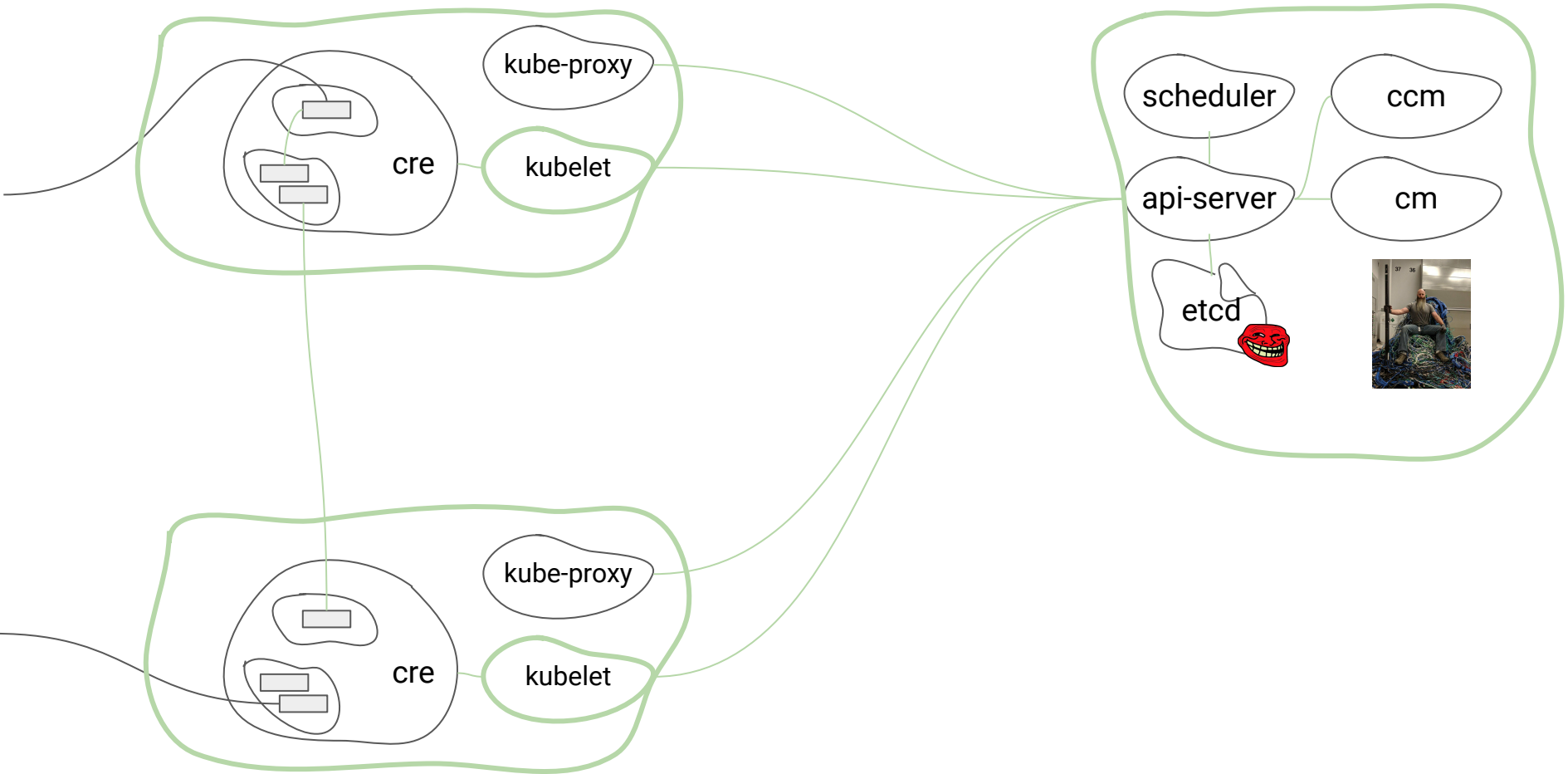
Istio mTLS

- Istio ControlPlane is taking care of certificate lifecycle
- mTLS means that both endpoints get verified
- TLS termination will happen in the envoy proxies
- mTLS communication is the default in Istio


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

KUBERMATIC

Encryption at Rest


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet

Encryption Config

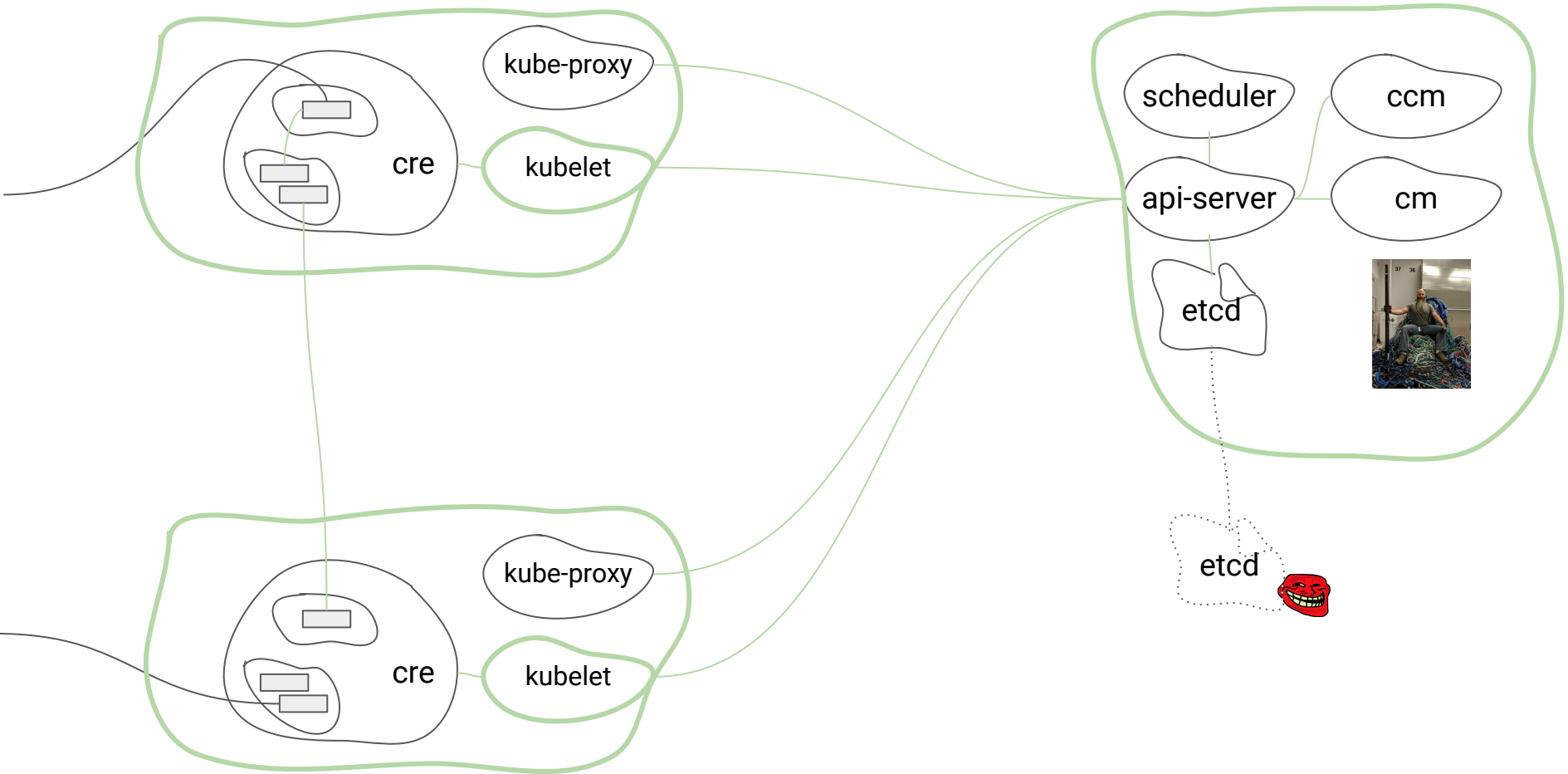
```yaml
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
  - resources:
      - secrets
    providers:
      - aescbc:
          keys:
            - name: key1
              secret: VcPcNPU5z75x15ZNsVMxp+NJqcnZ4SNK5375cBnbBjo=
      - identity: {}
```

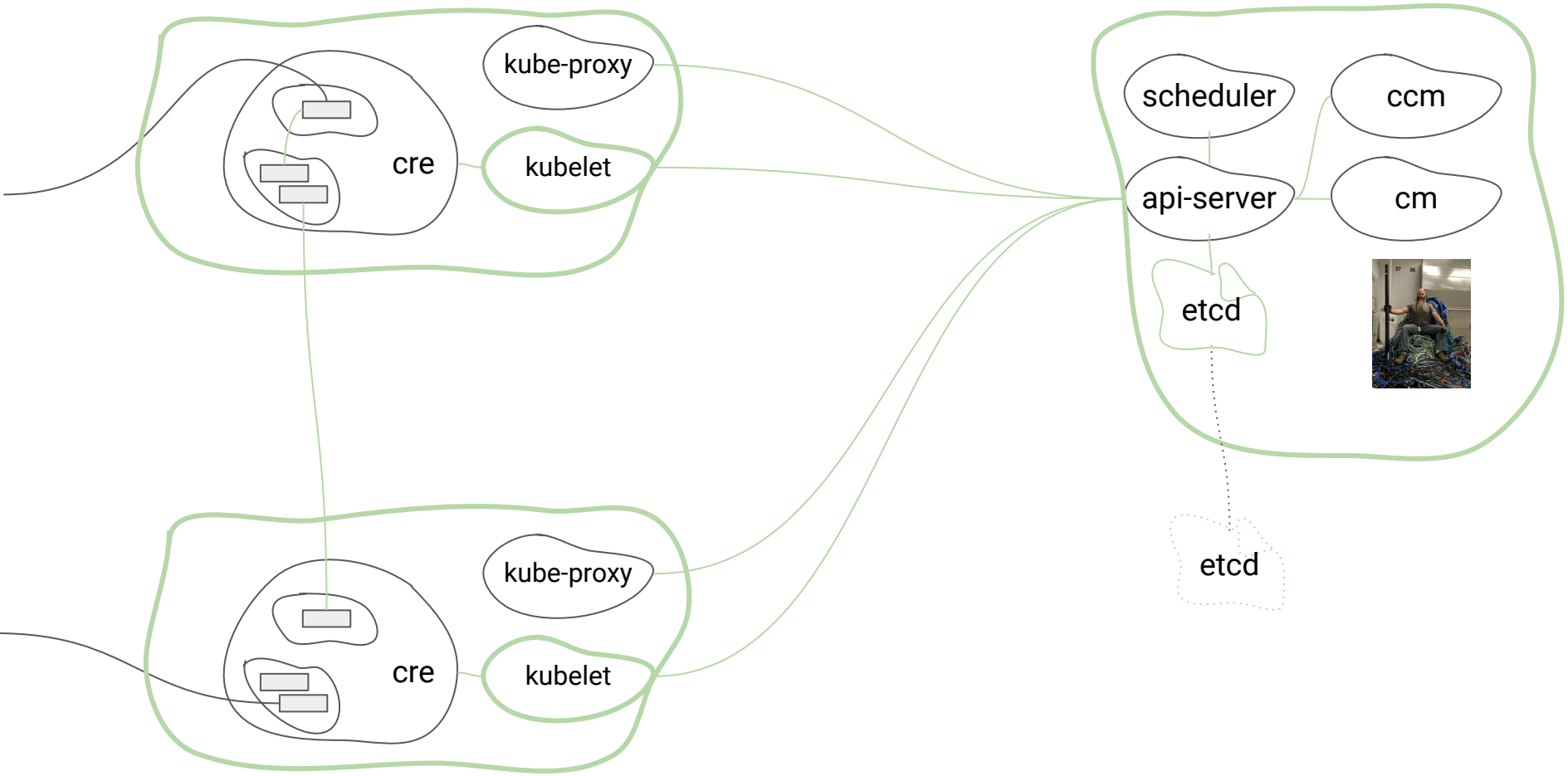
Teaching api-server Encryption Config

```
...
spec:
  containers:
  - command:
    - kube-apiserver
    - --encryption-provider-config=/apiserver/encryption-config.yaml
    - --advertise-address=167.235.74.251
    - --allow-privileged=true
...
```

Key	Value	revision
X	1	1
X	3	2
X	2	3
X	5	4

```
$> etcdctl get x          # returns 5

$> etcdctl get x --rev=0  # returns 5

$> etcdctl get x --rev=4  # returns 5

$> etcdctl get x --rev=1  # returns 1

$> etcdctl get x --rev=2  # returns 3

$> etcdctl get x --rev=3  # returns 2
```


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet
etcd

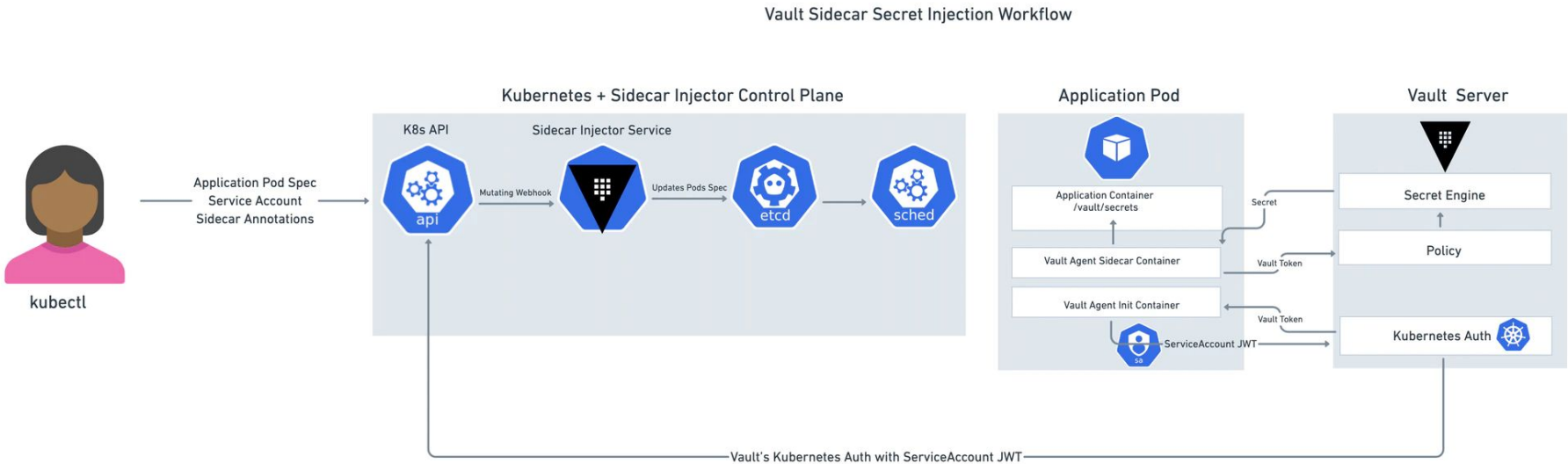

kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet
etcd

KUBERMATIC

What about Secrets?

KUBERMATIC

Hashicorp Vault


Vault Sidecar Secret Injection Workflow
Kubernetes + Sidecar Injector Control Plane
Application Pod
Vault Server
K8s API
Sidecar Injector Service
kubectl
Application Pod Spec
Service Account
Sidecar Annotations
api
Mutating Webhook
Updates Pods Spec
etcd
sched
Application Container
/vault/secrets
Vault Agent Sidecar Container
Vault Agent Init Container
Secret
Vault Token
Vault Token
ServiceAccount JWT
Secret Engine
Policy
Kubernetes Auth
Vault's Kubernetes Auth with ServiceAccount JWT

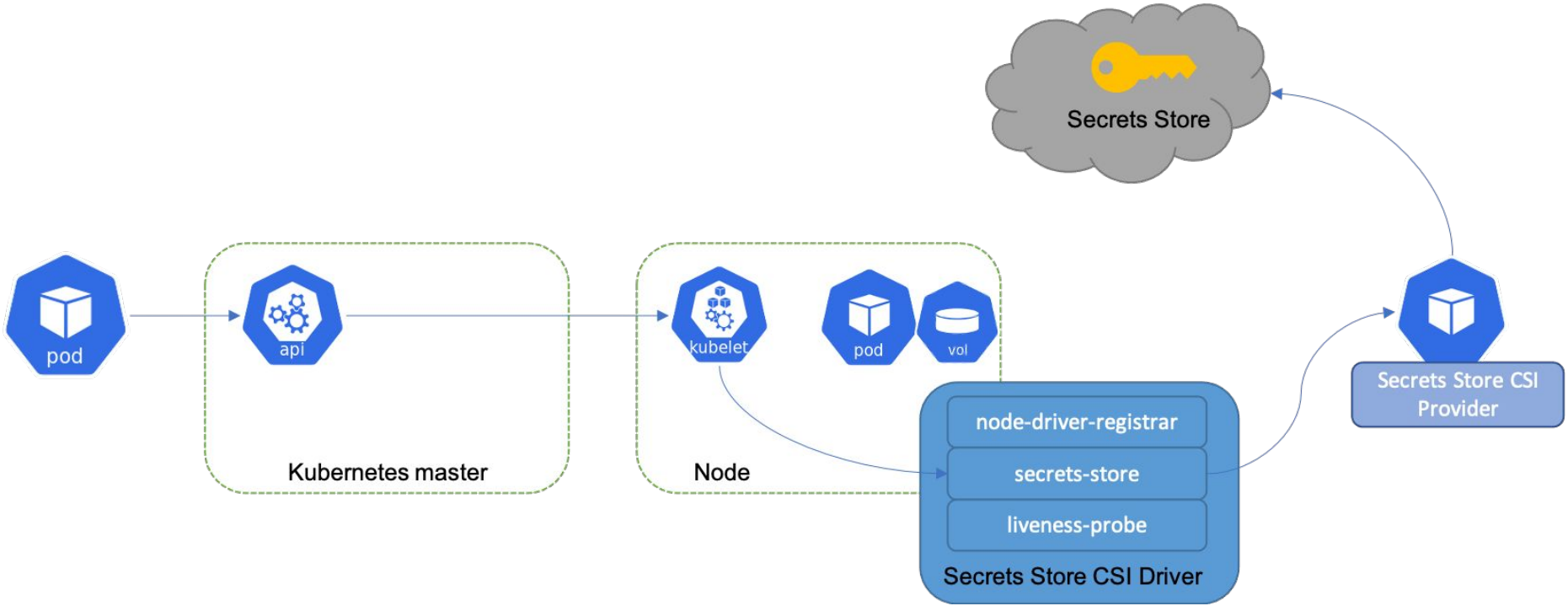
KUBERMATIC

Container Storage Interface

Container Storage Interface

- Volume plugins used to be “in-tree”
- CSI allows third-party storage providers to implement their storage solutions on their own pace (not bound to the Kubernetes release process)

Secrets Store CSI Driver



Secret Provider Class

```yaml
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: my-provider
spec:
  provider: gcp
  parameters:
...
```

Usage in Pods

```
...
volumes:
  - name: secrets-store-inline
    csi:
      driver: secrets-store.csi.k8s.io
      readOnly: true
      volumeAttributes:
        secretProviderClass: "my-provider"
```

Secrets Store CSI Driver

- Integrates secrets stores with Kubernetes via a CSI (Container Storage Interface) volume.
- Support for
  - AWS
  - Azure
  - GCP
  - Vault

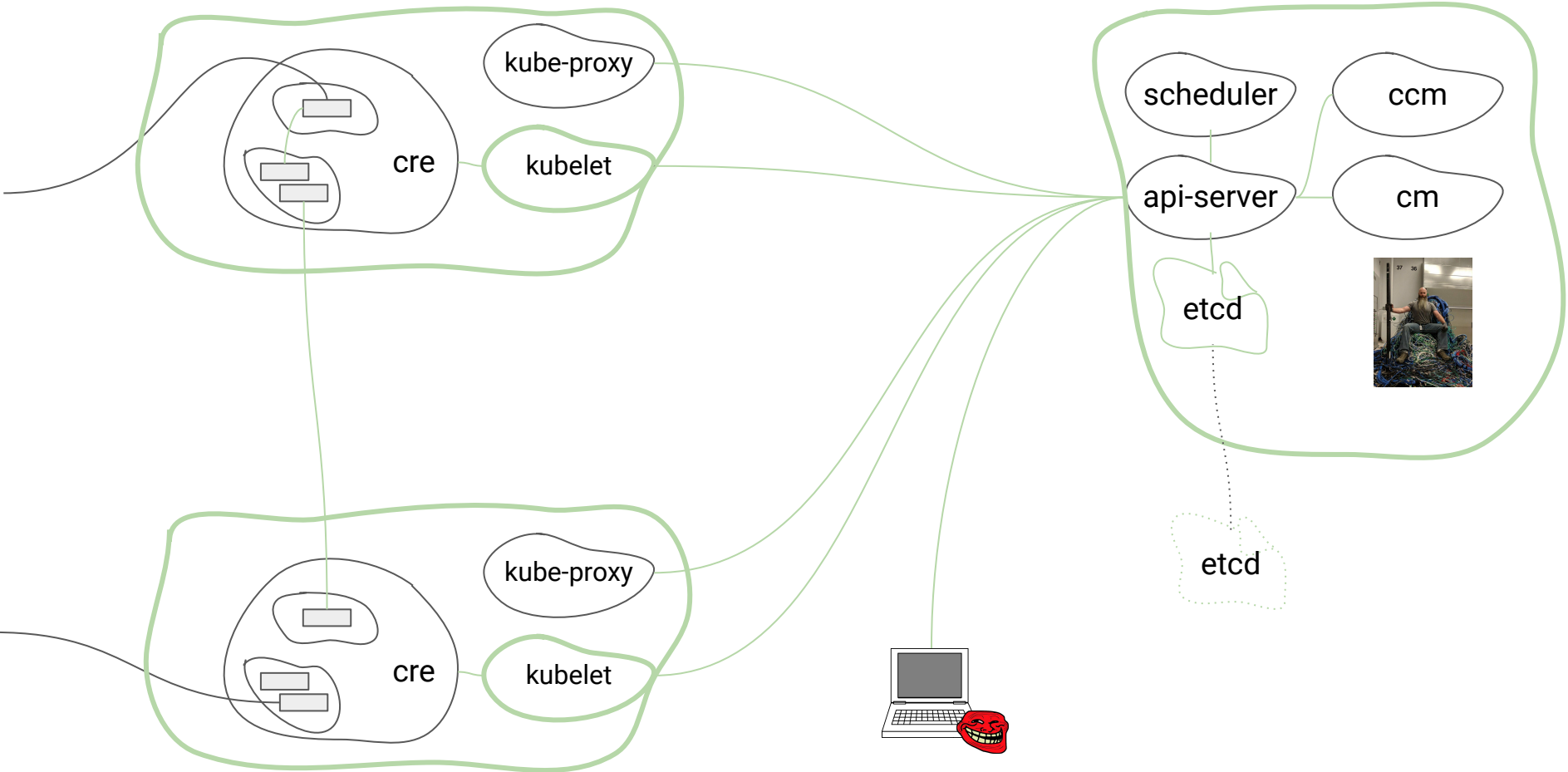
KUBERMATIC

King Admin checking the craftsmen

Static Pod Analysis & Image Scanning

KUBERMATIC

Static Pod Analysis


kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
etcd
kube-proxy
cre
kubelet
etcd


```
[
  {
    "object": "Pod/security-context-demo.default",
    "valid": true,
    "message": "Failed with a score of -30 points",
    "score": -30,
    "scoring": {
      "critical": [
        {
          "selector": "containers[] .securityContext .capabilities .add == SYS_ADMIN",
          "reason": "CAP_SYS_ADMIN is the most privileged capability and should always be avoided"
        }
      ],
      "advise": [
        {
          "selector": "containers[] .securityContext .runAsNonRoot == true",
          "reason": "Force the running image to run as a non-root user to ensure least privilege"
        },
        {
          // ...
        }
      ]
    }
  }
]
```

Do static Pod analysis within your CI/CD pipeline

Do static Pod analysis via CronJob in your Prod Environment

KUBERMATIC

Image Scanning


```
> trivy image --ignore-unfixed nginx
2023-03-08T07:39:00.907+0100    INFO    Vulnerability scanning is enabled
2023-03-08T07:39:00.907+0100    INFO    Secret scanning is enabled
2023-03-08T07:39:00.907+0100    INFO    If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2023-03-08T07:39:00.907+0100    INFO    Please see also https://aquasecurity.github.io/trivy/v0.38/docs/secret/scanning/#recommendation for faster secret detection
2023-03-08T07:39:00.928+0100    INFO    Detected OS: debian
2023-03-08T07:39:00.928+0100    INFO    Detecting Debian vulnerabilities...
2023-03-08T07:39:00.962+0100    INFO    Number of language-specific files: 0

nginx (debian 11.6)

Total: 13 (UNKNOWN: 0, LOW: 0, MEDIUM: 10, HIGH: 3, CRITICAL: 0)
```

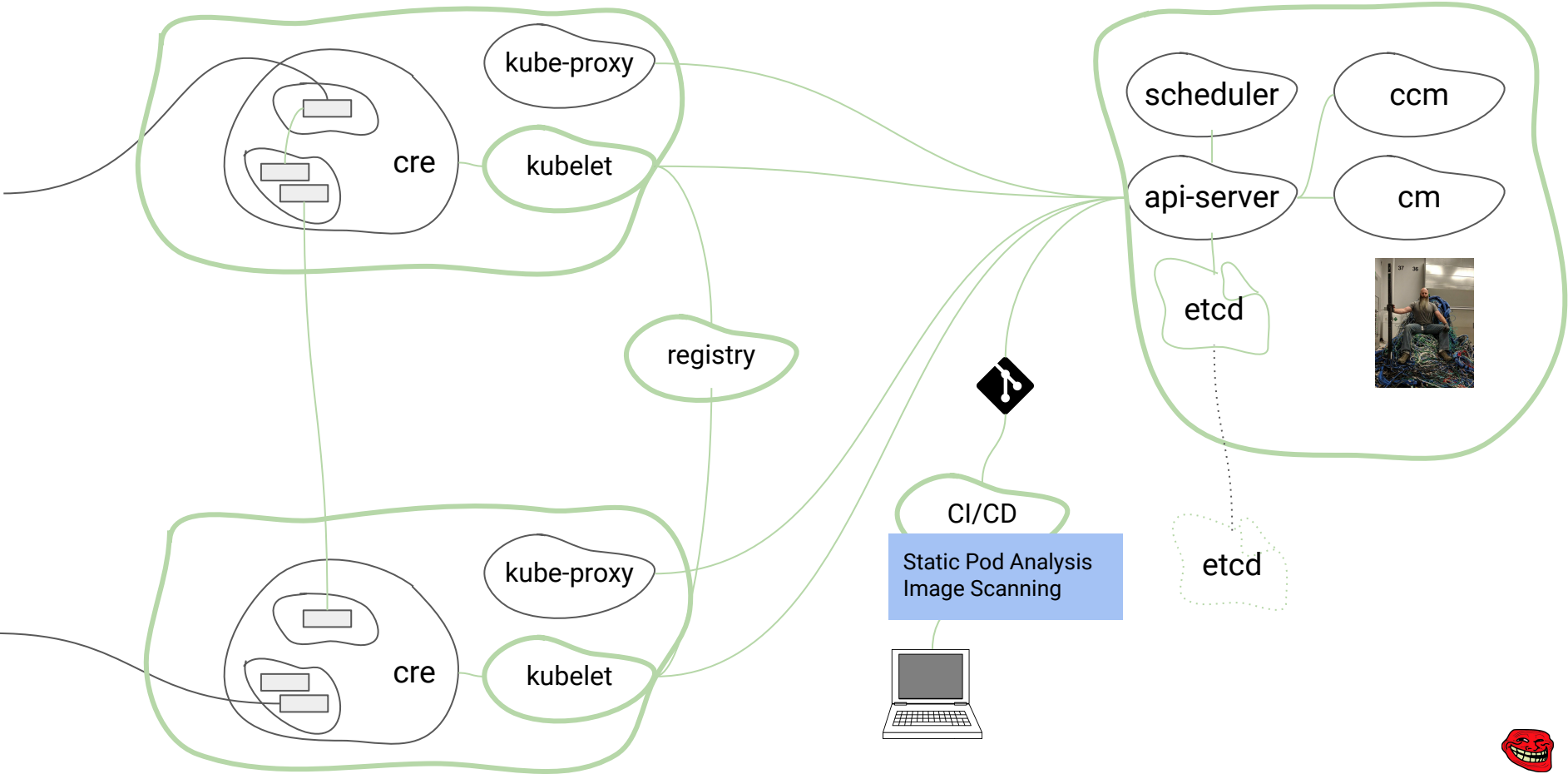
Library	Vulnerability	Severity	Installed Version	Fixed Version	Title
curl	CVE-2023-23916	HIGH	7.74.0-1.3+deb11u5	7.74.0-1.3+deb11u7	curl: HTTP multi-header compression denial of service https://avd.aquasec.com/nvd/cve-2023-23916
libcurl4					
libgnutls30	CVE-2023-0361		3.7.1-5+deb11u2	3.7.1-5+deb11u3	gnutls: timing side-channel in the TLS RSA key exchange code https://avd.aquasec.com/nvd/cve-2023-0361
libtiff5	CVE-2023-0795	MEDIUM	4.2.0-1+deb11u3	4.2.0-1+deb11u4	libtiff: out-of-bounds read in extractContigSamplesShifted16bits() in tools/tiffcrop.c https://avd.aquasec.com/nvd/cve-2023-0795

```
> trivy image --severity CRITICAL nginx
2023-03-08T07:39:22.673+0100    INFO    Vulnerability scanning is enabled
2023-03-08T07:39:22.673+0100    INFO    Secret scanning is enabled
2023-03-08T07:39:22.673+0100    INFO    If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2023-03-08T07:39:22.673+0100    INFO    Please see also https://aquasecurity.github.io/trivy/v0.38/docs/secret/scanning/#recommendation for faster secret detection
2023-03-08T07:39:22.704+0100    INFO    Detected OS: debian
2023-03-08T07:39:22.704+0100    INFO    Detecting Debian vulnerabilities...
2023-03-08T07:39:22.742+0100    INFO    Number of language-specific files: 0

nginx (debian 11.6)

Total: 3 (CRITICAL: 3)
```

Library	Vulnerability	Severity	Installed Version	Fixed Version	Title
curl	CVE-2023-23914	CRITICAL	7.74.0-1.3+deb11u5		curl: HSTS ignored on multiple requests https://avd.aquasec.com/nvd/cve-2023-23914
libcurl4					
libdb5.3	CVE-2019-8457		5.3.28+dfsg1-0.8		sqlite: heap out-of-bound read in function rtreenode() https://avd.aquasec.com/nvd/cve-2019-8457

Do Image scanning within your CI/CD pipeline

Do Image Scanning via CronJob in your Prod Environment

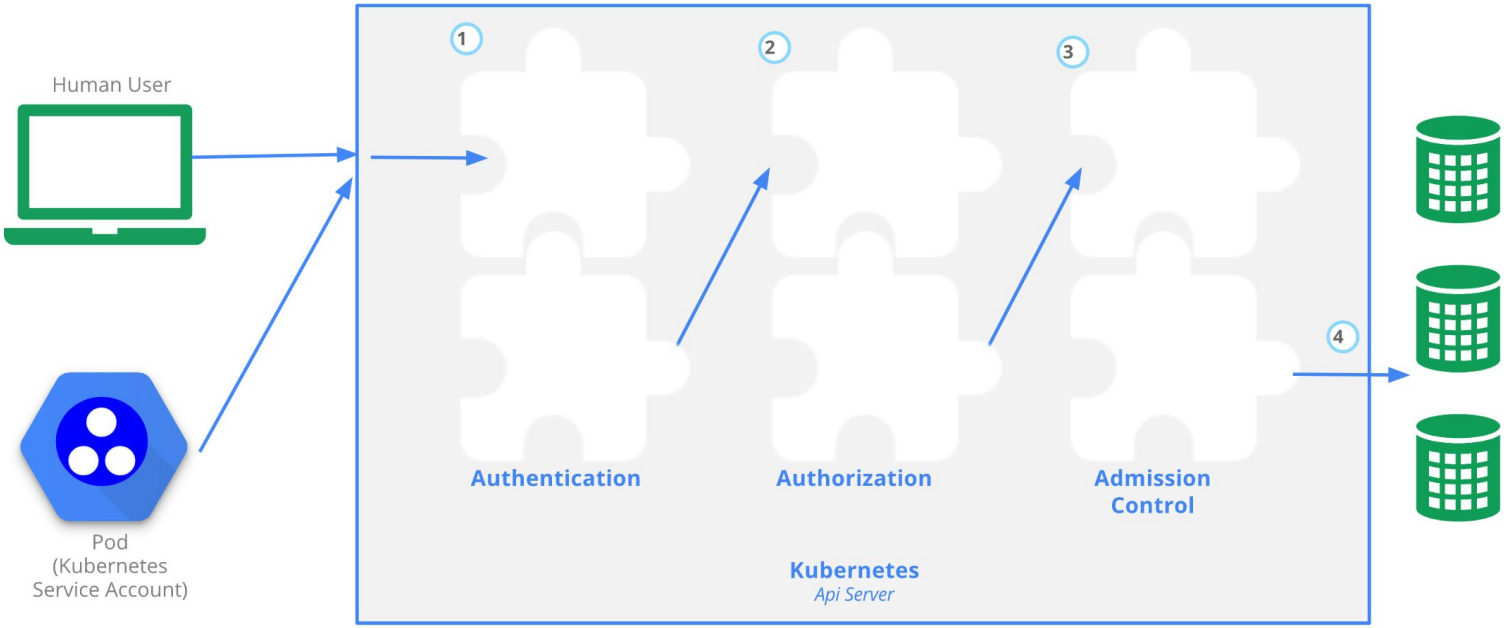

kube-proxy
cre
kubelet
registry
scheduler
ccm
api-server
cm
etcd
CI/CD
Static Pod Analysis
Image Scanning
etcd
kube-proxy
cre
kubelet

KUBERMATIC

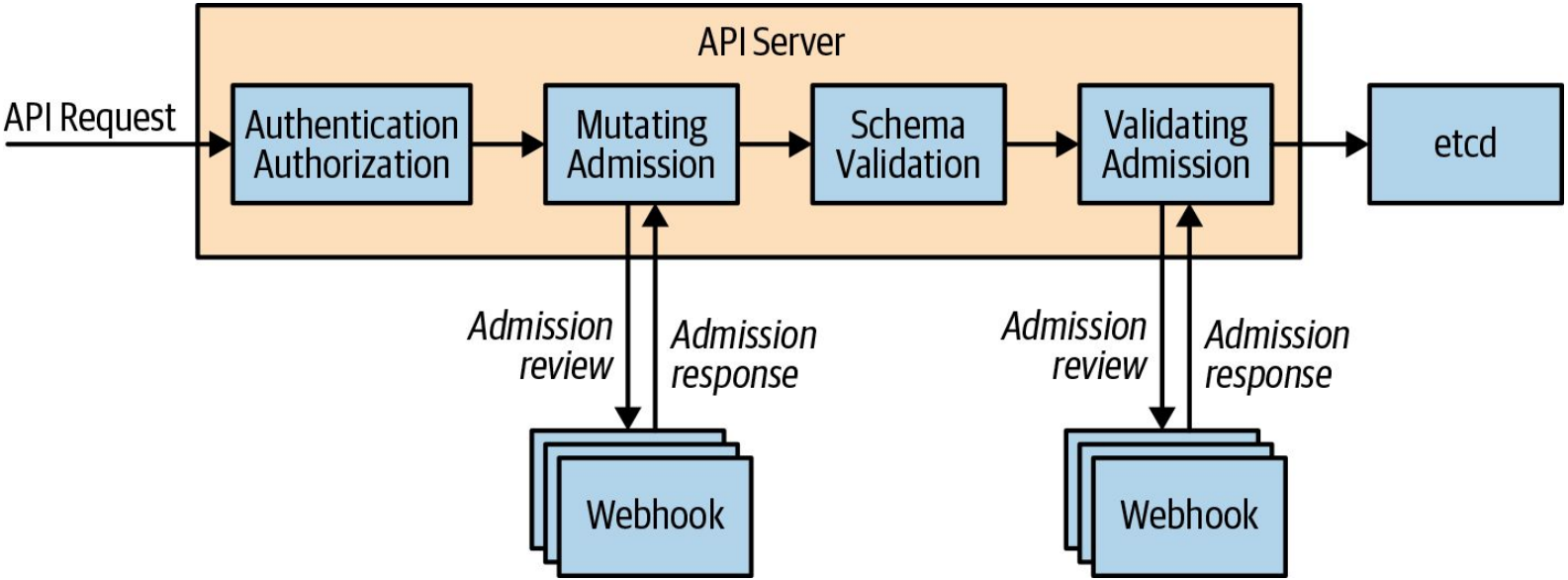
The god Kyverno backs King Admin

Restricting the workloads

api-server


Human User
Pod
(Kubernetes
Service Account)
1
2
3
4
Authentication
Authorization
Admission
Control
Kubernetes
Api Server

Webhooks


API Server
API Request
Authentication
Authorization
Mutating
Admission
Schema
Validation
Validating
Admission
etcd
Admission
review
Admission
response
Admission
review
Admission
response
Webhook
Webhook

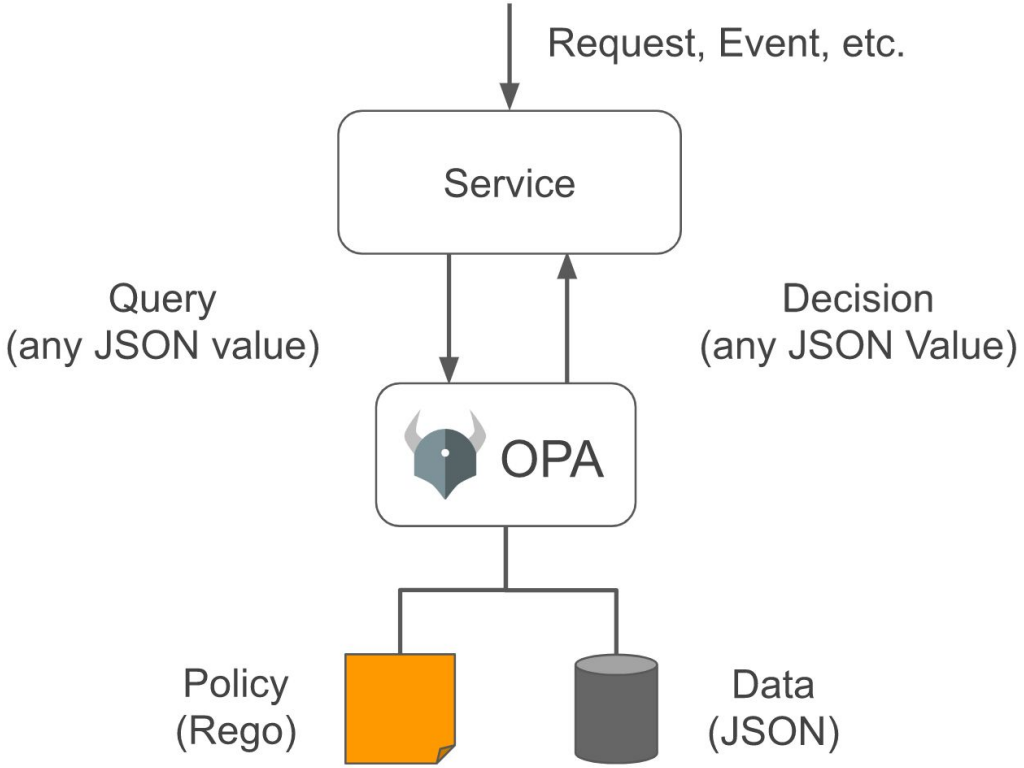
PodSecurityPolicy

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
```

> **Removed feature**
>
> PodSecurityPolicy was deprecated in Kubernetes v1.21, and removed from Kubernetes in v1.25.

```yaml
  volumes: ['*']
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  runAsUser:
    rule: 'RunAsAny'
...
```

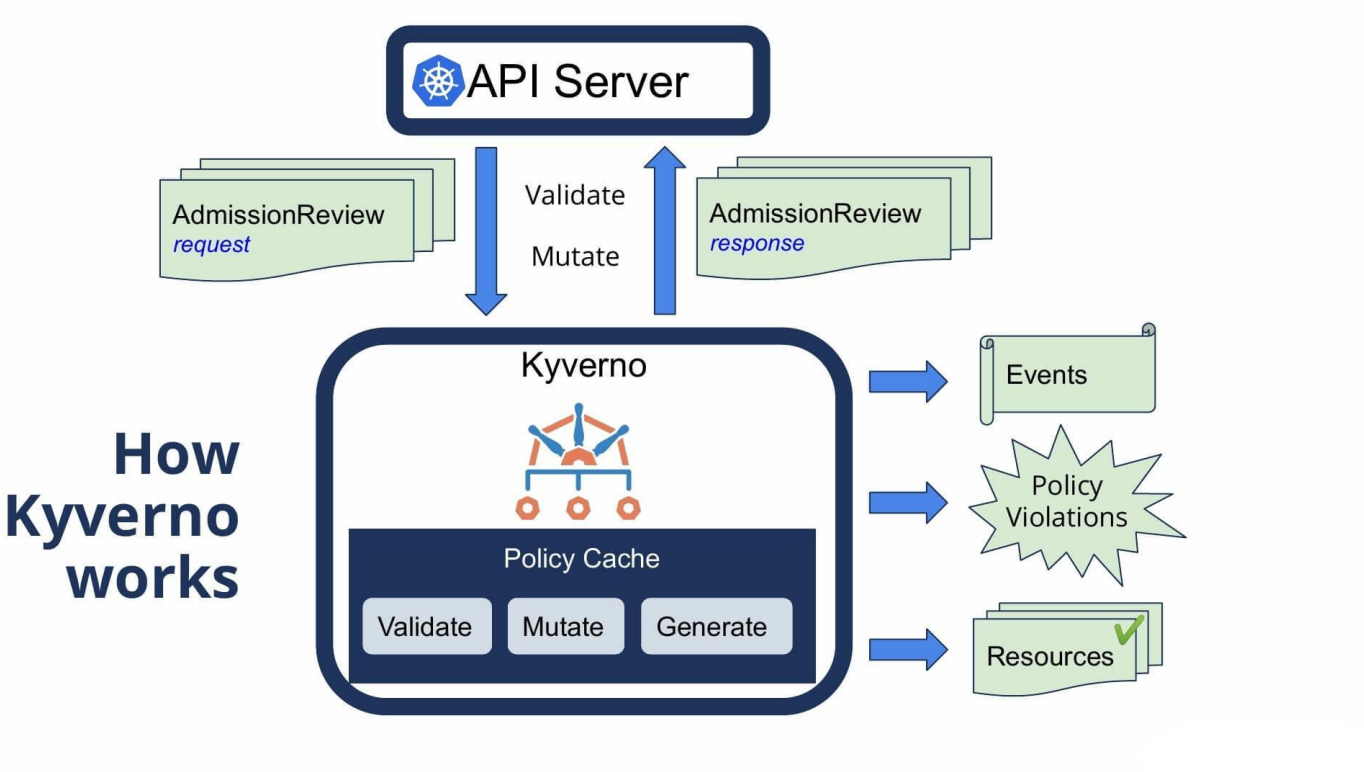
OPA


Request, Event, etc.
Service
Query
(any JSON value)
Decision
(any JSON Value)
OPA
Policy
(Rego)
Data
(JSON)

Rego

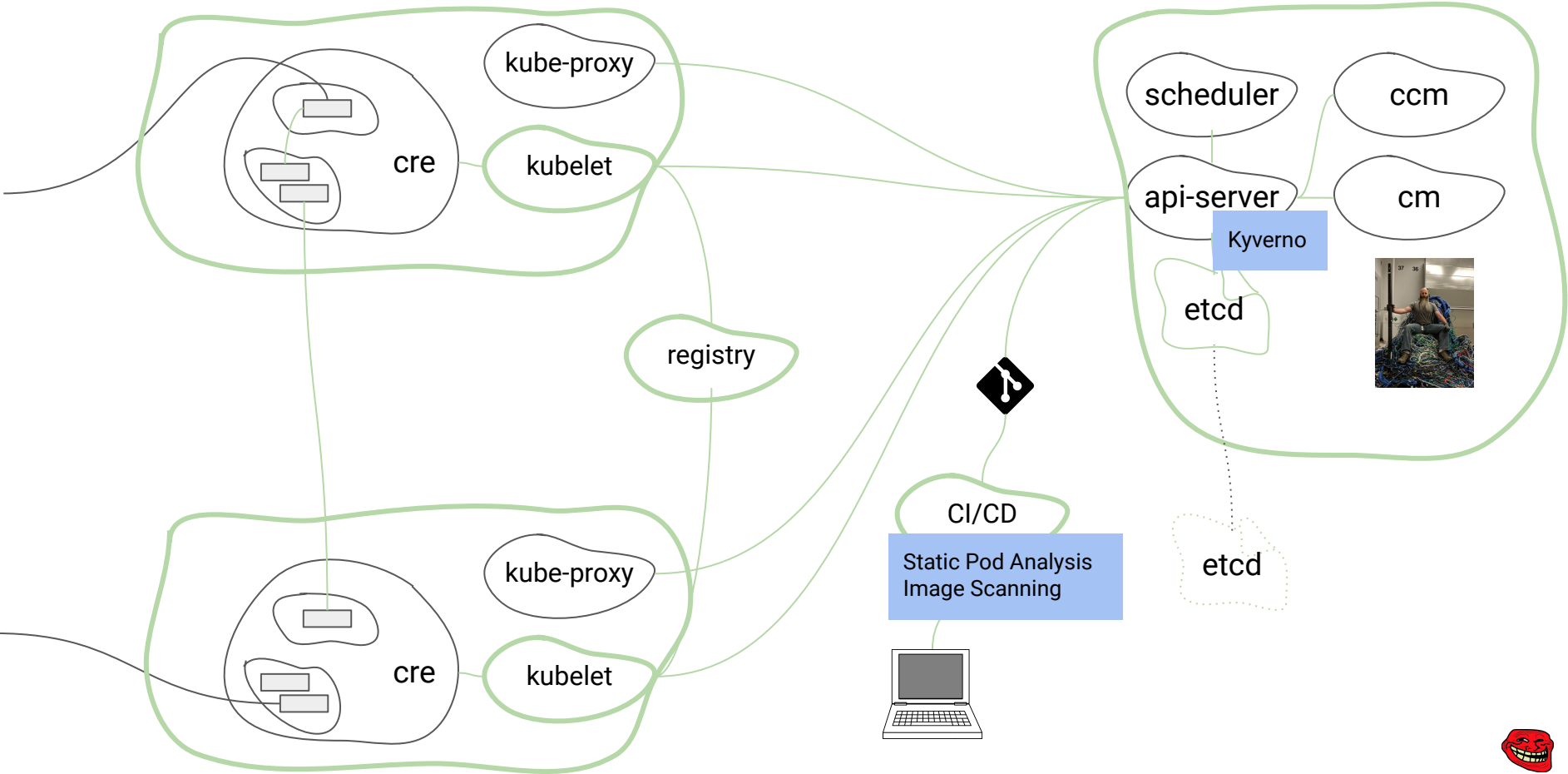
```
package kubernetes.admission

deny[msg] {
    input.request.kind.kind == "Pod"
    image := input.request.object.spec.containers[_].image
    not startswith(image, "my-image-registry.com/")
    msg := sprintf("image '%v' comes from untrusted registry", [image])
}
```

Kyverno


How Kyverno works
API Server
AdmissionReview
request
Validate
Mutate
AdmissionReview
response
Kyverno
Policy Cache
Validate
Mutate
Generate
Events
Policy Violations
Resources

Kyverno ClusterPolicy

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-image-registries
  annotations:
    policies.kyverno.io/title: Restrict Image Registries
spec:
  validationFailureAction: enforce
  rules:
    - name: validate-registries
      match:
        resources:
          kinds:
            - Pod
      validate:
        message: "Unknown image registry."
        pattern:
          spec:
            containers:
              - image: "my-private-registry/*"
```


kube-proxy
cre
kubelet
registry
kube-proxy
cre
kubelet
scheduler
ccm
api-server
cm
Kyverno
etcd
CI/CD
Static Pod Analysis
Image Scanning
etcd

KUBERMATIC

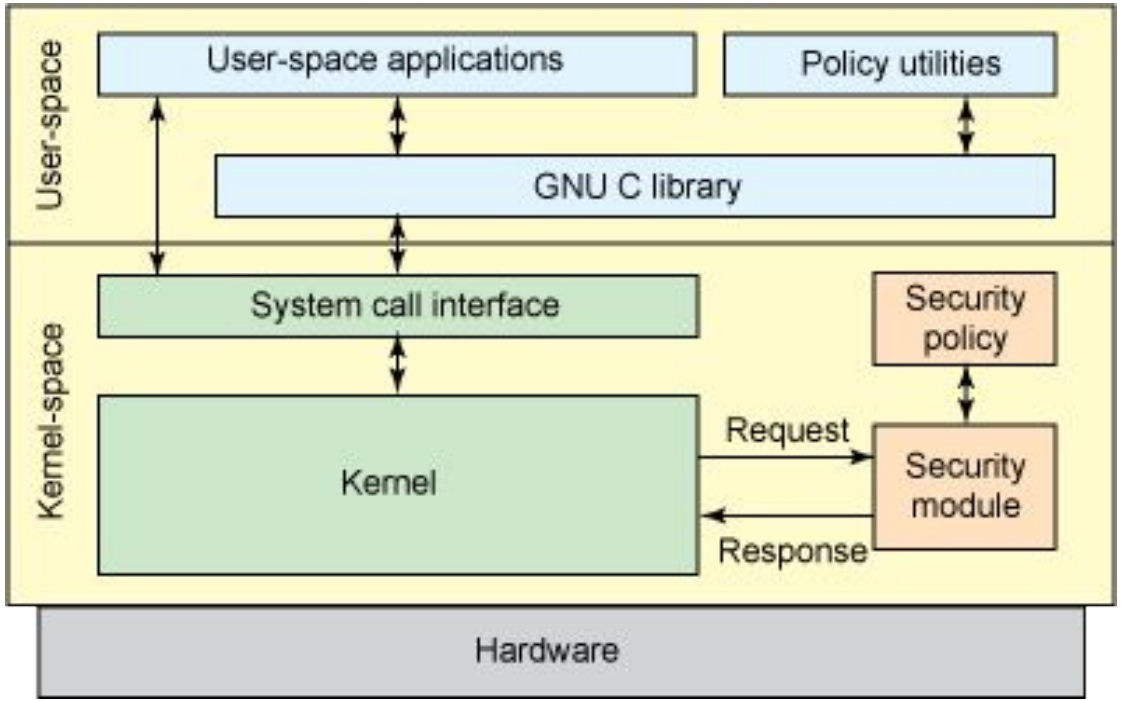
The Kernel gods back King Admin

Restricting containers via OS Host

AppArmor

AppArmor is a Mandatory Access Control framework that functions as an LSM (Linux Security Module). It is used to whitelist or blacklist a subject's (program's) access to an object (file, path, etc.).

Linux Security Module (LSM)



AppArmor Profile

```
#include <tunables/global>

profile my-apparmor-profile flags=(attach_disconnected) {
  #include <abstractions/base>
  file,
  deny /** w,
}
```

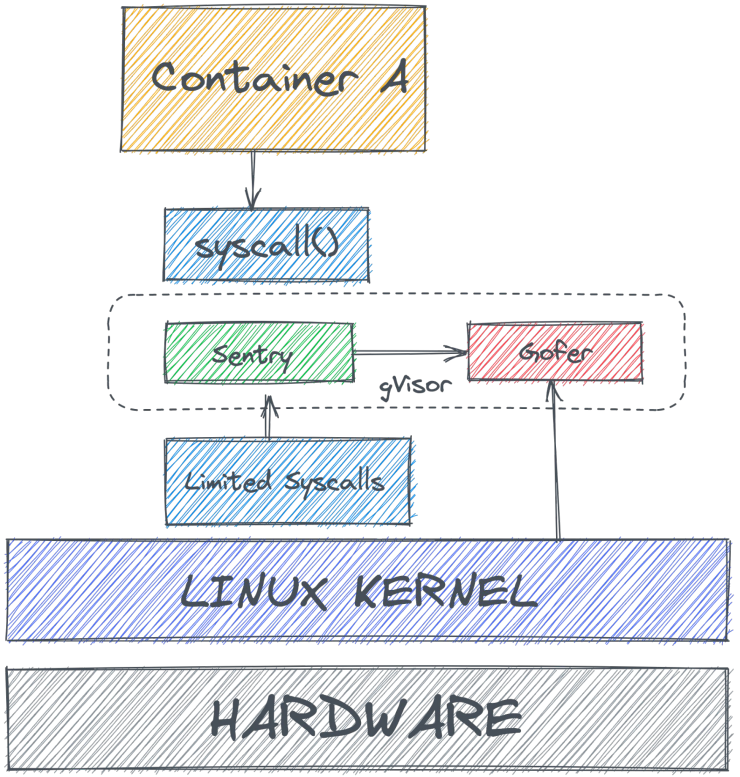
Engage AppArmor Profile

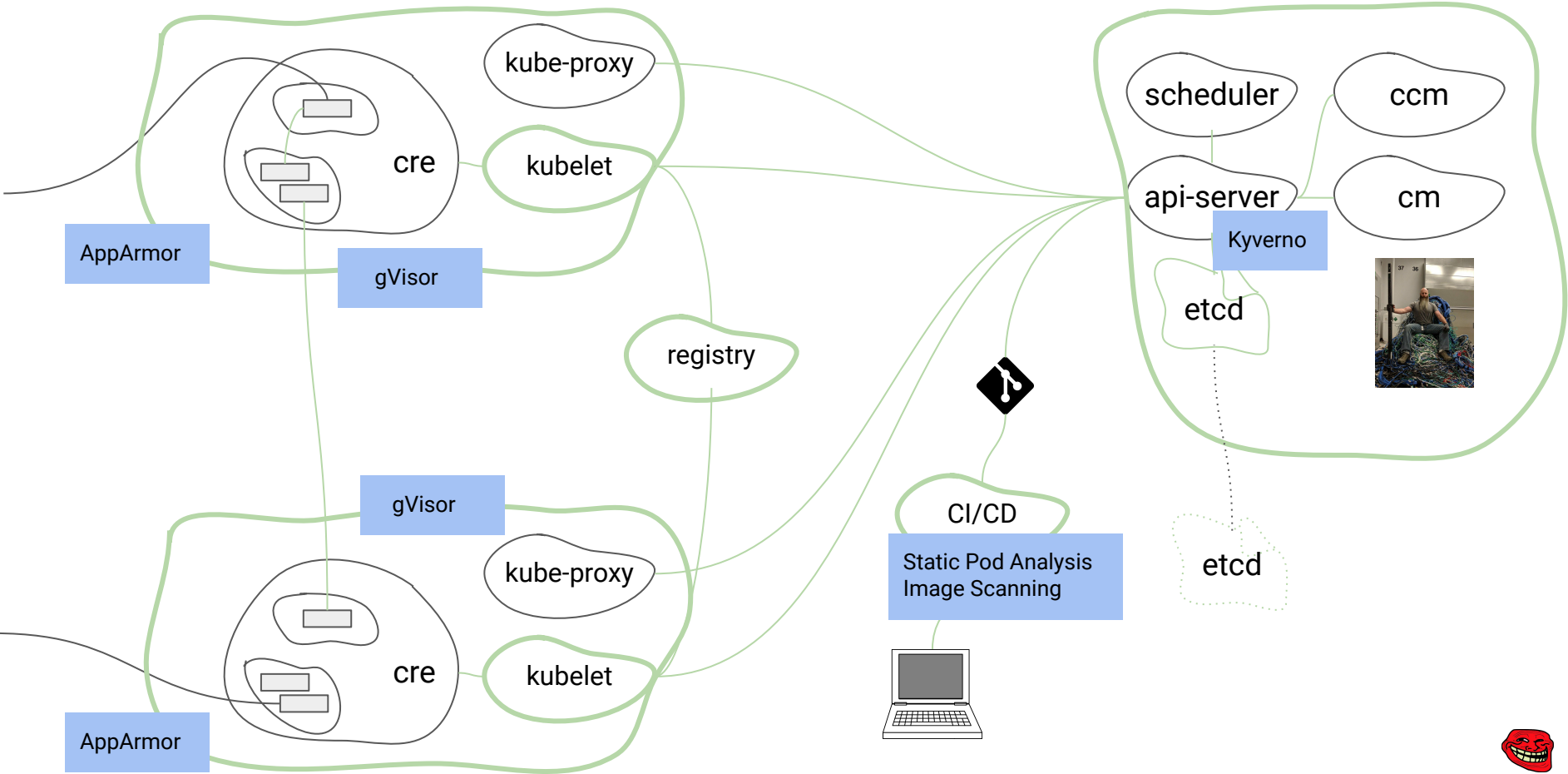
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-suboptimal-pod
  annotations:
    container.apparmor.security.beta.kubernetes.io/my-ubuntu: localhost/my-apparmor-profile
...
```

Sandbox Runtime gVisor

A user-space kernel for containers.

- Another layer of separation, between the container and the kernel.
- Simulates kernel syscalls with limited functionality.
- Runs in the user-space
- Runtime is called runsc




kube-proxy
cre
kubelet
AppArmor
gVisor
registry
scheduler
ccm
api-server
cm
Kyverno
etcd
gVisor
kube-proxy
cre
kubelet
AppArmor
CI/CD
Static Pod Analysis
Image Scanning
etcd

KUBERMATIC

King Admins monitoring I

Kubernetes Auditing

Audit Policy

```yaml
# audit-policy.yaml
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages: ["RequestReceived"]
rules:
  - namespaces: ["default"]
    verbs: ["get", "list", "watch"]
    resources:
      - groups: ""
        resources: ["secrets"]
    level: Request
```

Stages

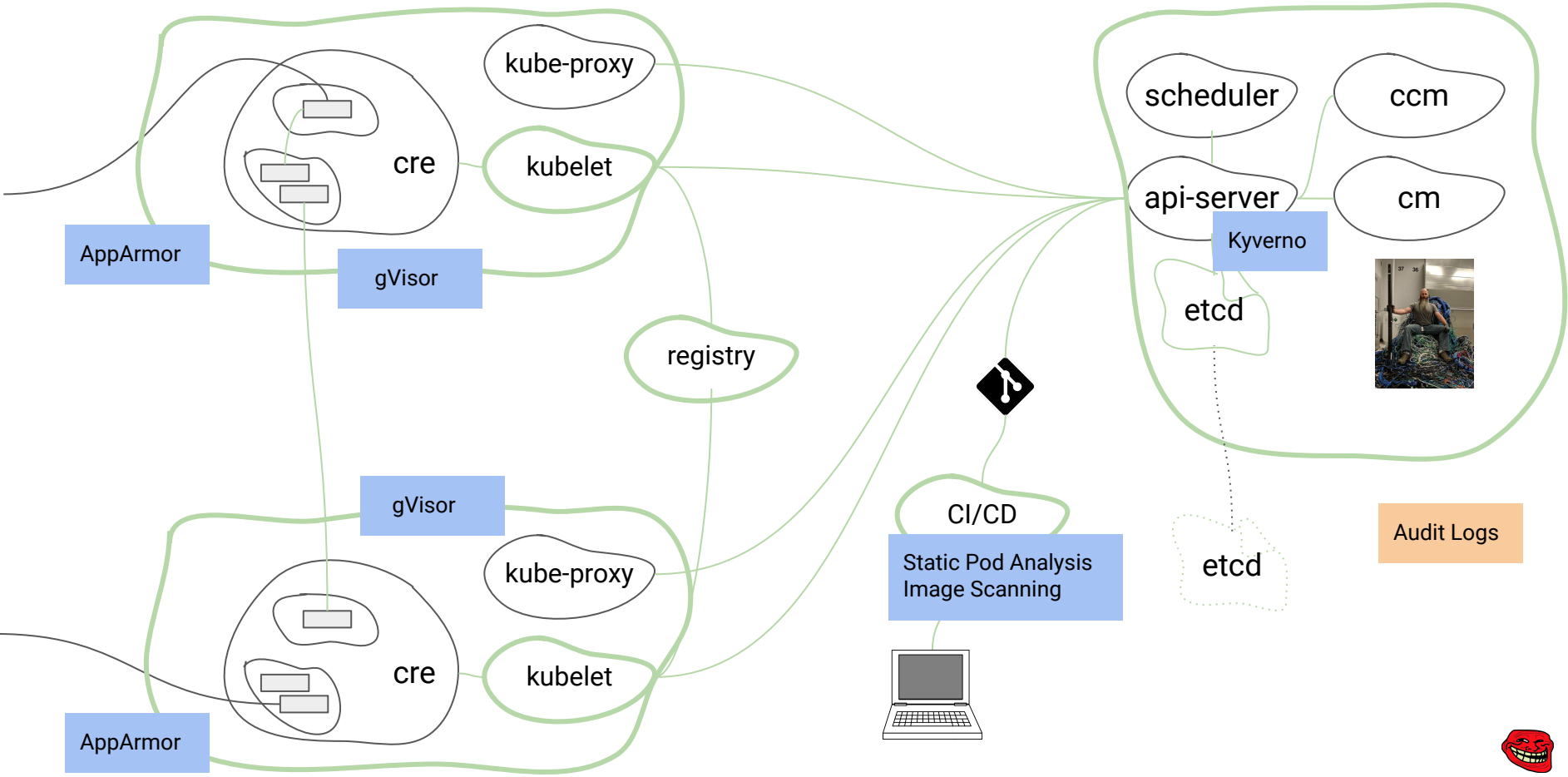
- **RequestReceived**
  The stage for events generated as soon as the audit handler receives the request.
- **ResponseStarted**
  Once the response headers are sent, but before the response body is sent. This stage is only generated for long-running requests (e.g. watch).
- **ResponseComplete**
  Once the response body has been completed.
- **Panic**
  Events generated when a panic occurred.

Levels

- **None**
  Don’t log events that match this rule.
- **Metadata**
  Log request metadata (requesting user, timestamp, resource, verb, etc.) but not request or response body
- **Request**
  Log event metadata and request body but not response body.
- **RequestResponse**
  Log event metadata, request and response bodies.

Enabling Audits in api-server

```
--audit-policy-file=/apiserver/audit-policy.yaml
--audit-log-path=/apiserver/default-secrets.log
--audit-log-maxage=10
--audit-log-maxsize=100
```

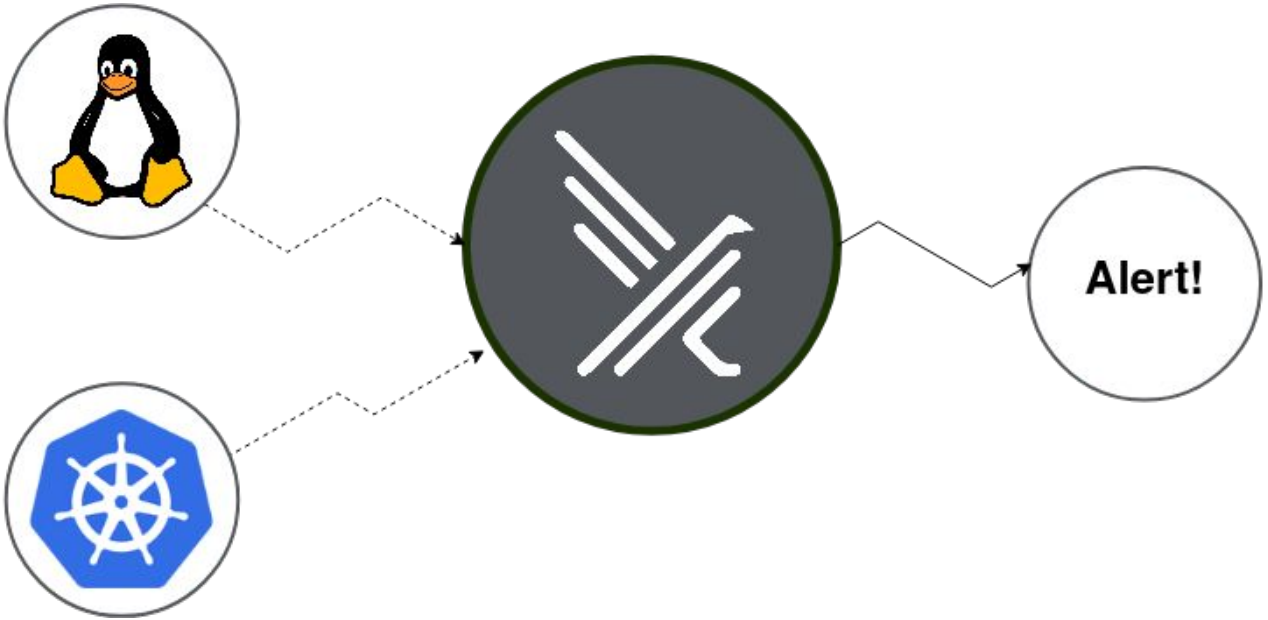

kube-proxy
cre
kubelet
AppArmor
gVisor
registry
scheduler
ccm
api-server
cm
Kyverno
etcd
gVisor
kube-proxy
cre
kubelet
AppArmor
CI/CD
Static Pod Analysis
Image Scanning
etcd
Audit Logs

KUBERMATIC

King Admins monitoring II

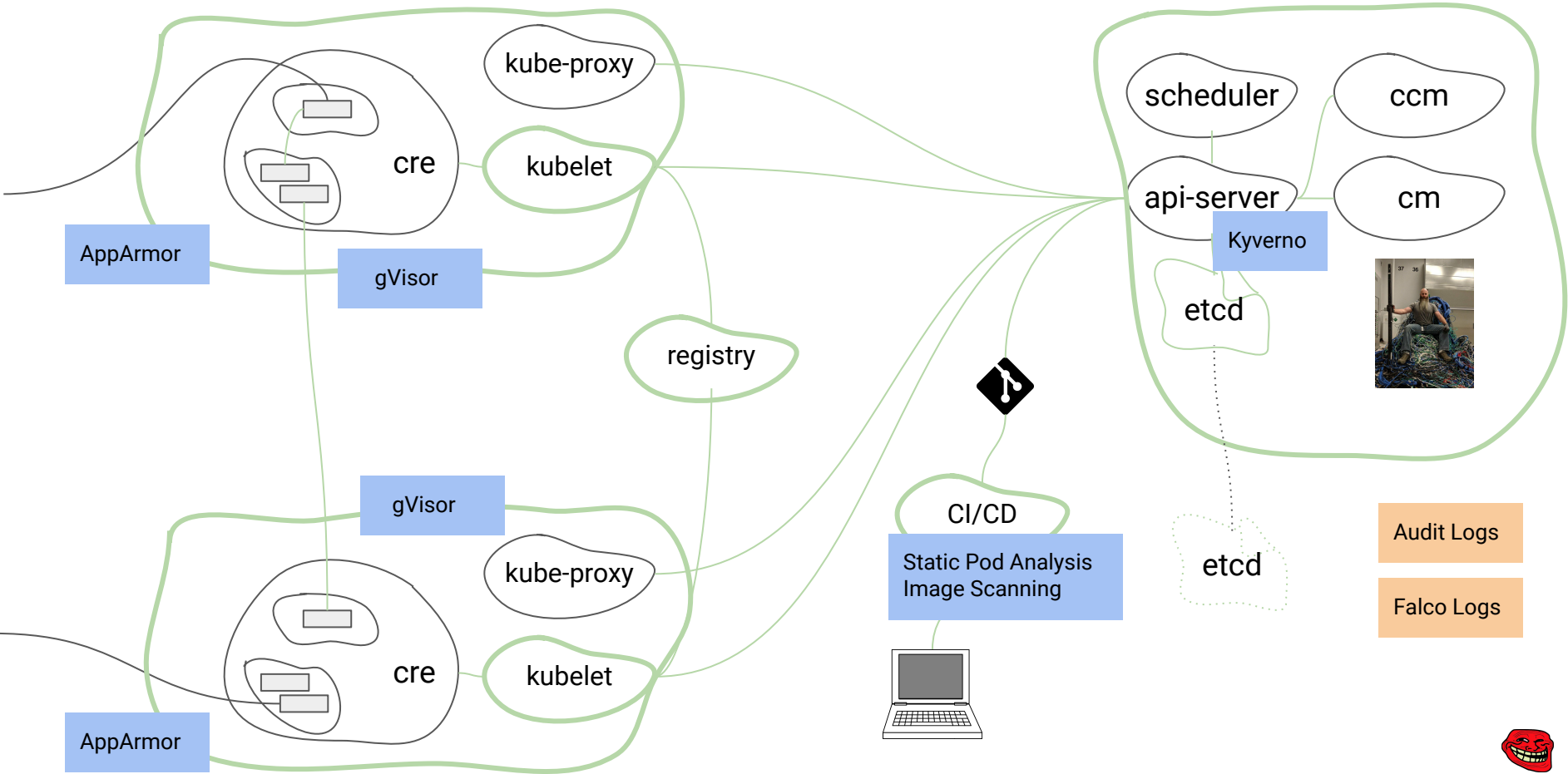
Using Falco for threat detection

Falco is a Kubernetes threat detection engine



Falco rule

```
- rule: Terminal shell in container
  desc: A shell was used as the entrypoint/exec point into a container with an attached terminal.
  condition: >
    spawned_process and container
    and shell_procs and proc.tty != 0
    and container_entrypoint
    and not user_expected_terminal_shell_in_container_conditions
  output: >
    A shell was spawned in a container with an attached terminal (user=%user.name user_loginuid=%user.loginuid
%container.info shell=%proc.name parent=%proc.pname cmdline=%proc.cmdline terminal=%proc.tty
container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [container, shell, mitre_execution]
```


kube-proxy
cre
kubelet
AppArmor
gVisor
registry
scheduler
ccm
api-server
cm
Kyverno
etcd
gVisor
kube-proxy
cre
kubelet
AppArmor
CI/CD
Static Pod Analysis
Image Scanning
etcd
Audit Logs
Falco Logs

KUBERMATIC

King Admin does benchmarking

Use kube-bench for benchmarking your cluster

CIS Center for Internet Security®
Creating Confidence in the Connected World.

CIS Ha

Home > CIS Benchmarks > **CIS Kubernetes Benchmarks**

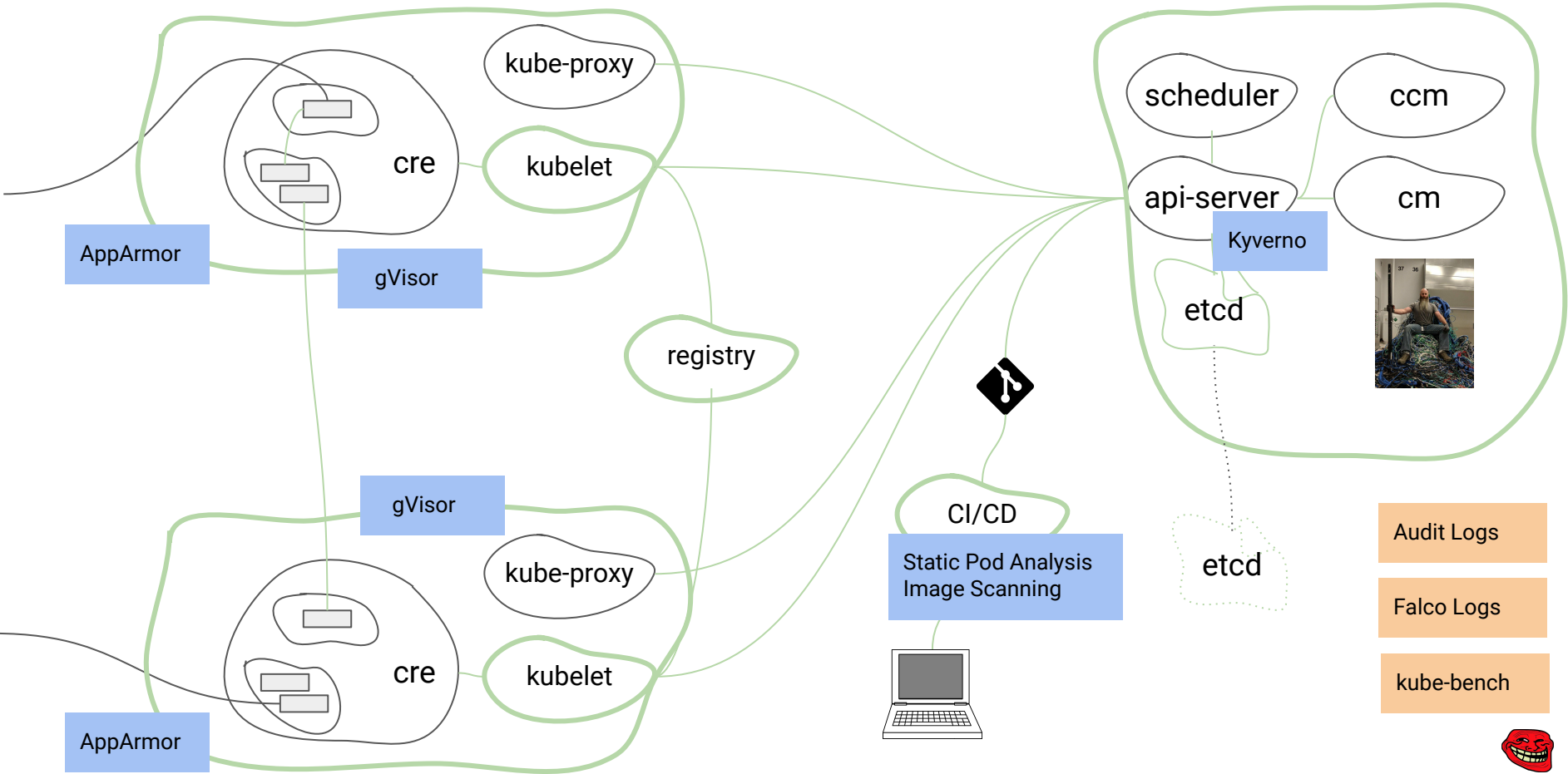
Securing Kubernetes

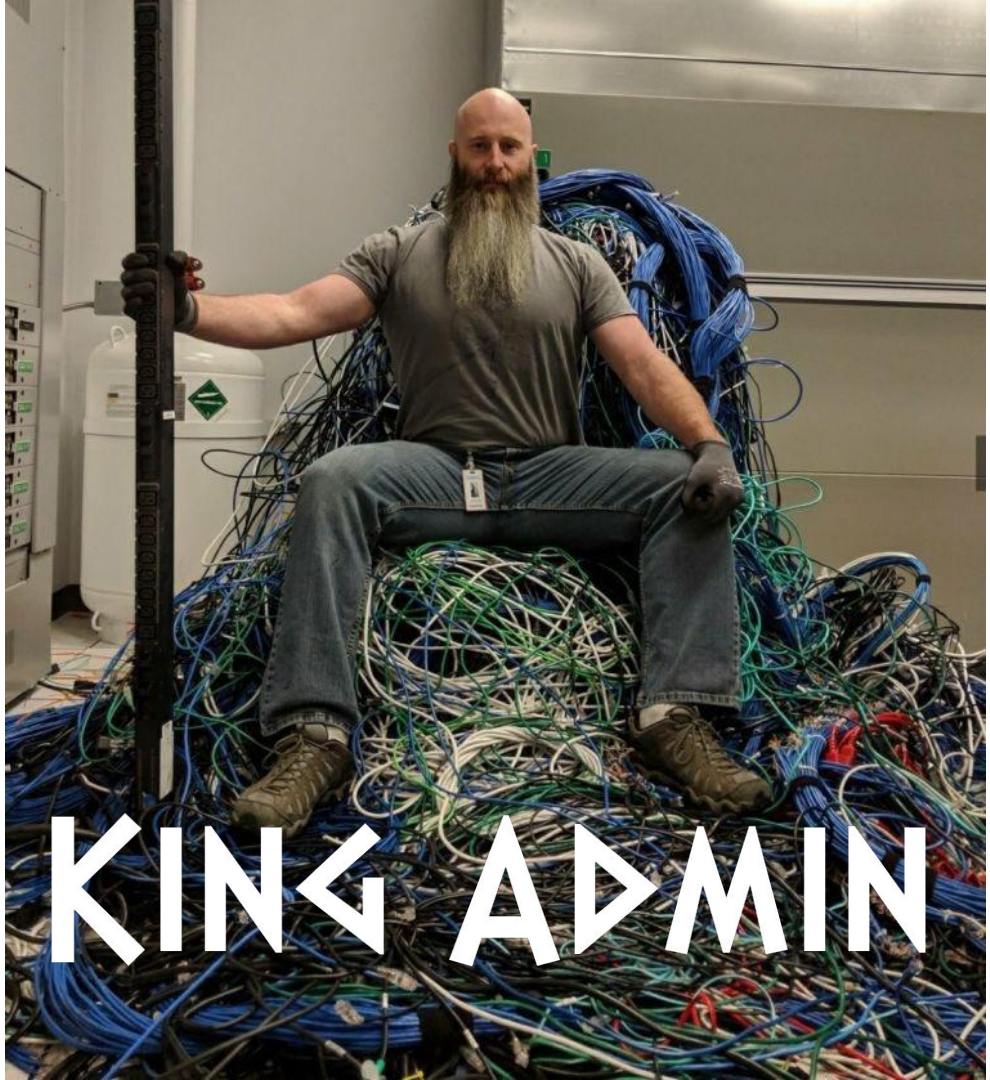
An objective, consensus-driven security guideline for the Kubernetes Server Software.

A step-by-step checklist to secure Kubernetes:

DOWNLOAD LATEST CIS BENCHMARK
FREE TO EVERYONE →

aqua
kube-bench


kube-proxy
cre
kubelet
AppArmor
gVisor
registry
scheduler
ccm
api-server
cm
Kyverno
etcd
gVisor
kube-proxy
cre
kubelet
AppArmor
CI/CD
Static Pod Analysis
Image Scanning
etcd
Audit Logs
Falco Logs
kube-bench


KING ADMIN

Summary I

- Firewalls!!!
- Authentication and Authorization config of kubelet
- Least privilege principle & RBAC
- Make Containers ReadOnly (whenever possible)
- Do not mount ServiceAccount Token (whenever possible)
- Encrypt sensitive data in Git Repos
- Make use of external Secret Store
- Use a ServiceMesh for Service to Service Encryption

Summary II

- Encrypt sensitive information in ETCD
- Ensure no plaintext secrets in ETCD backups
- Ensure your backups are on a safe place
- Rotate the ETCD encryption key
- Make use of small images
- Do not store sensitive information into your Images
- Do static Pod Analysis
- Do Image Scanning
- Manifest static Pod Analysis and Image Scanning into your CI/CD pipeline

Summary III

- Make use of Kyverno Admission Webhook
- Make use of Host Level Security tools like Apparmor
- Do Audits via enabling Audit Logging
- Do Threat Detection via Falco
- Benchmark your cluster via eg kube-bench

Honorable Mentions

- Make use of small images / distroless images
- Make your Cotainers read-only (readOnlyRootFilesystem in SecuritContext)
- ImagePolicyWebhooks
- Network Policies: Firewalls within your cluster
- Container Sandboxing: Gvisor or Kata Containers


I'll be back!!!

KUBERMATIC

Questions?

KUBERMATIC

Thank you!

koray@kubermatic.com

@korayoksay

koksay