KubeCon

CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon | CloudNativeCon

THE LINUX FOUNDATION OPEN SOURCE SUMMIT

AI_dev Open Source GenAI & ML Summit

China 2024

Ensuring Success with Kyverno in Production

What You Must Know

Shuting Zhao, Nirmata

About Me

Shuting Zhao

Nirmata

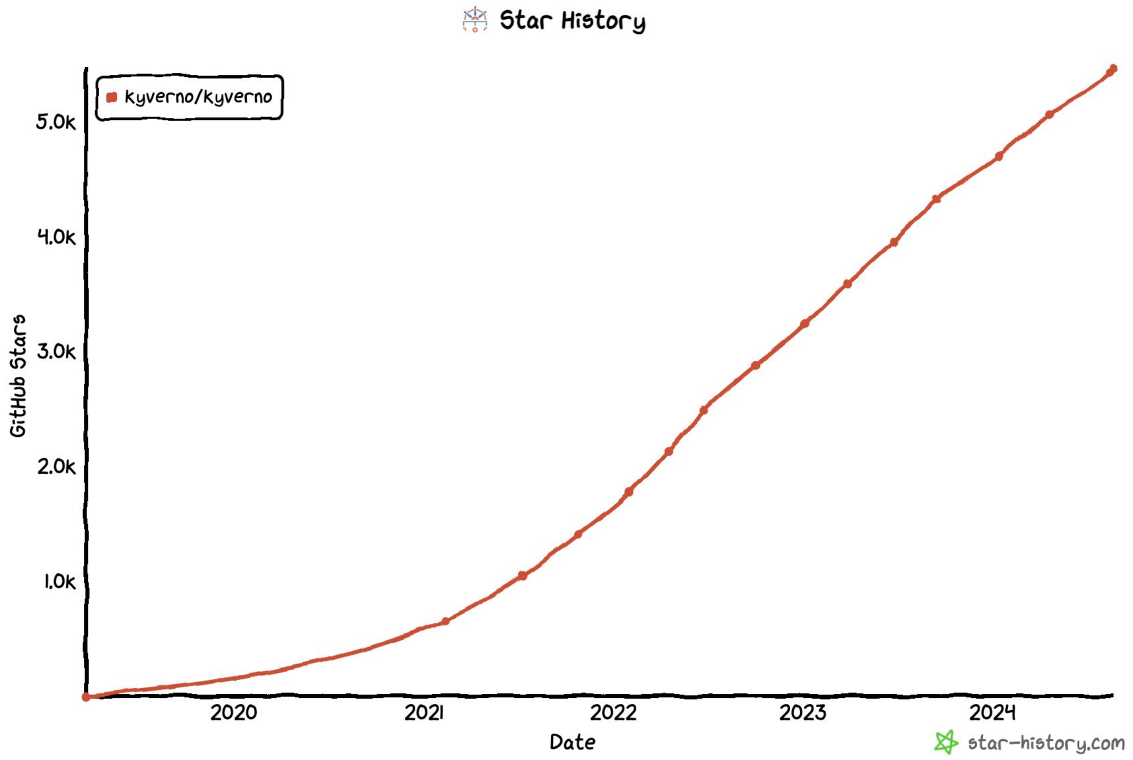
- Kyverno Maintainer
- Staff Engineer

in linkedin.com/in/shuting-zhao

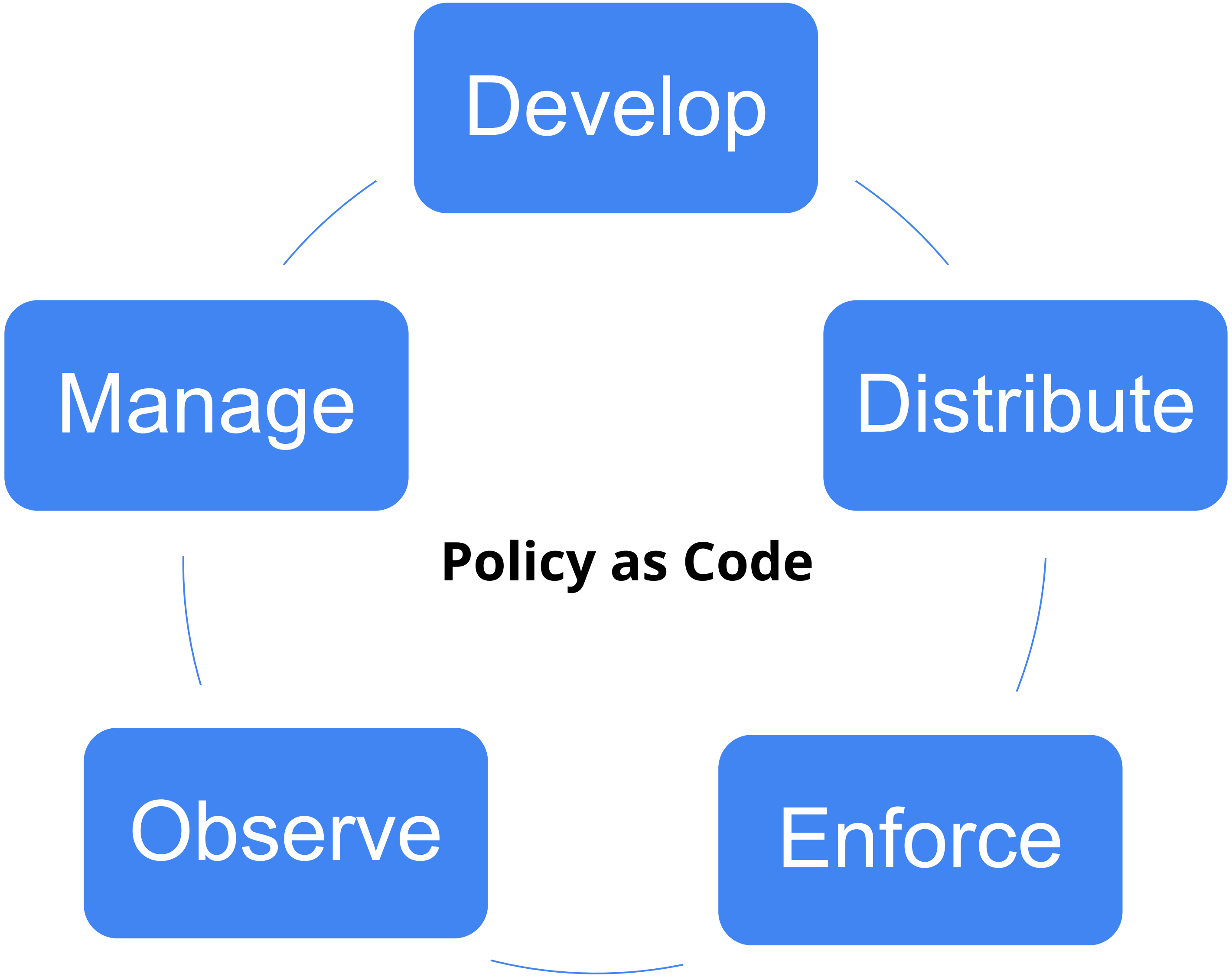
@ShutingZhao2

Kyverno

- Kyverno means “Govern” in Greek
- Cloud Native Policy Management
- CNCF incubating project
- Fast growing community
  - 5.5K+ GitHub Stars
  - 3100+ Slack members



Cloud Native Policy Management



Develop

Distribute

Enforce

Observe

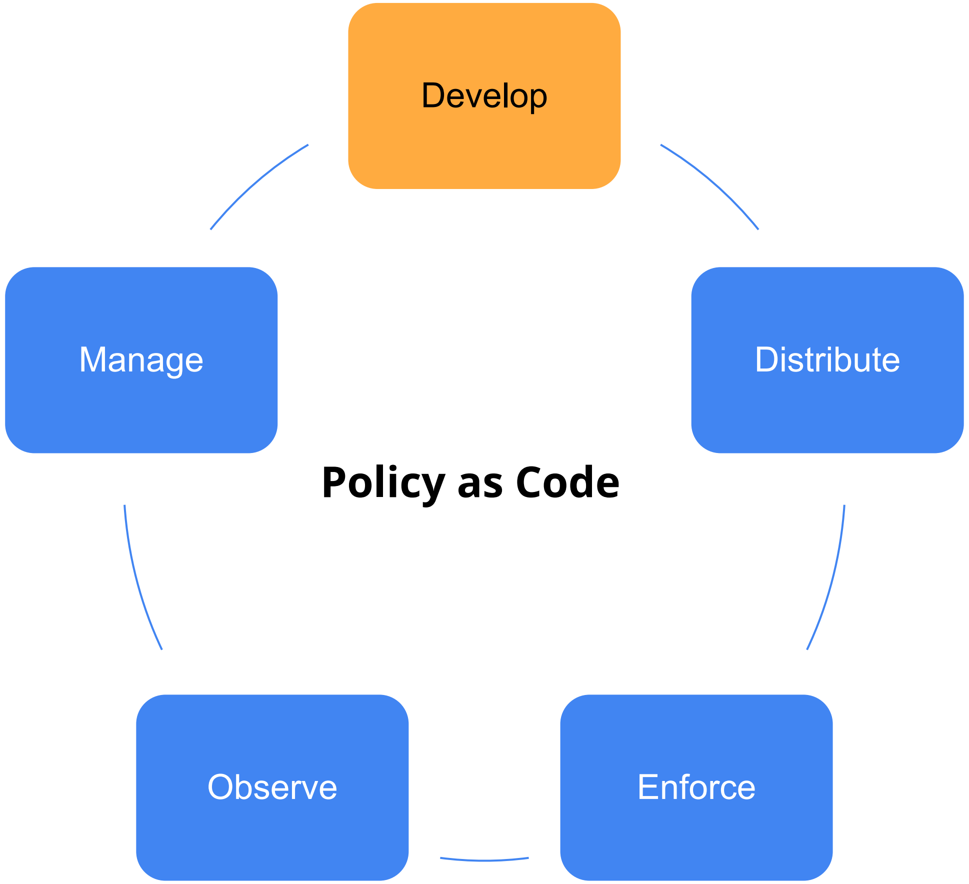
Manage

Policy as Code

Cloud Native Policy Management

Develop:

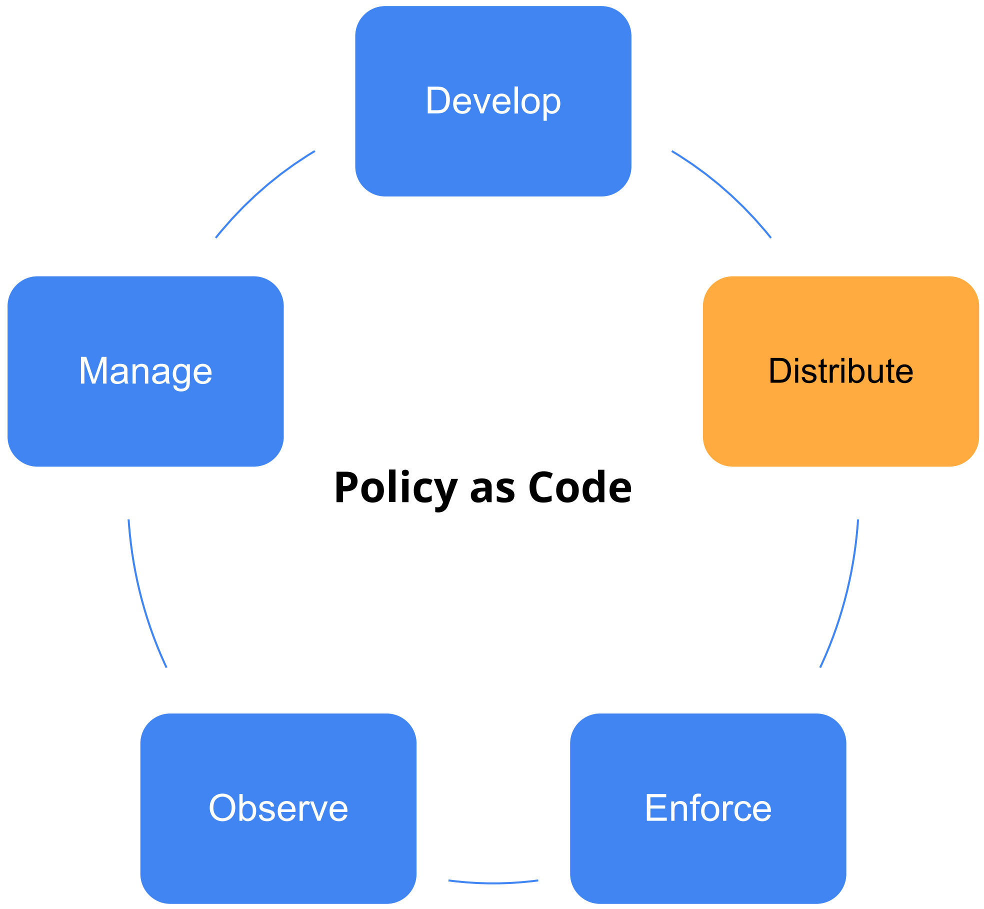
- Low code policies
- Easy to develop and test for Kubernetes admins and users
- Addresses use cases across validation, mutation, generation, cleanup, image verification
- JMESPath, CEL, and all features for complex logic



Cloud Native Policy Management

Distribute:

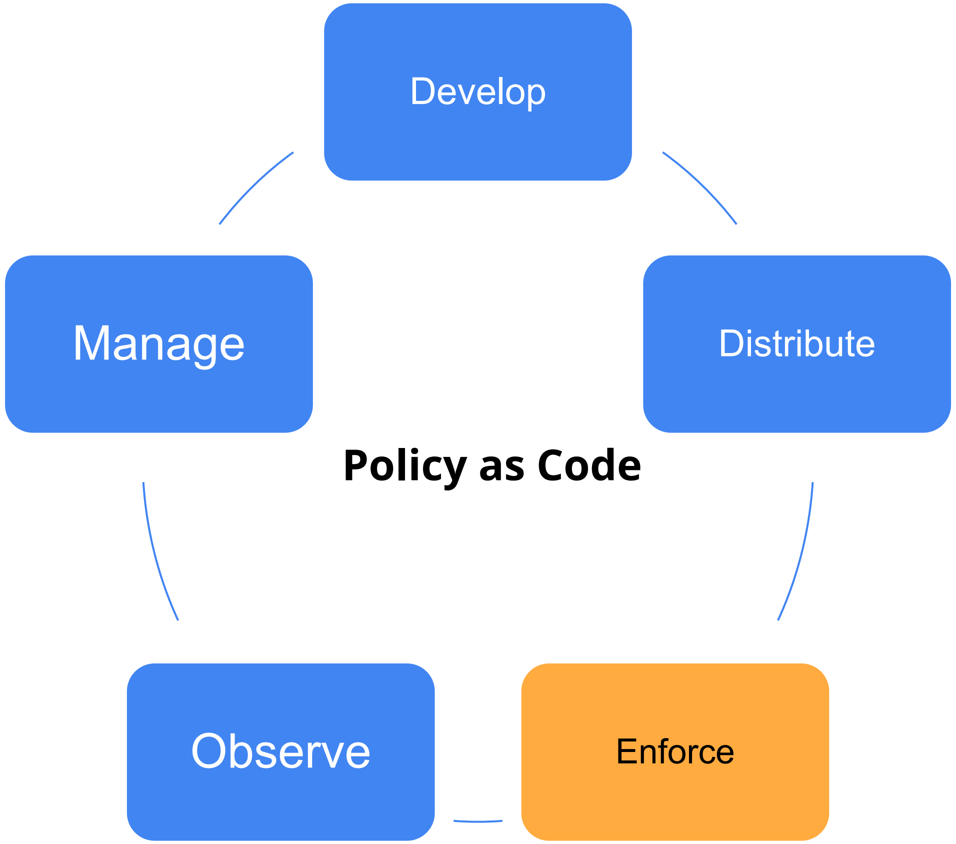
- Use Kubernetes APIs to deploy and manage
- Works with any Kubernetes management tool
- Use kubectl, Kustomize, etc.
- Use GitOps workflows



Cloud Native Policy Management

Enforce:

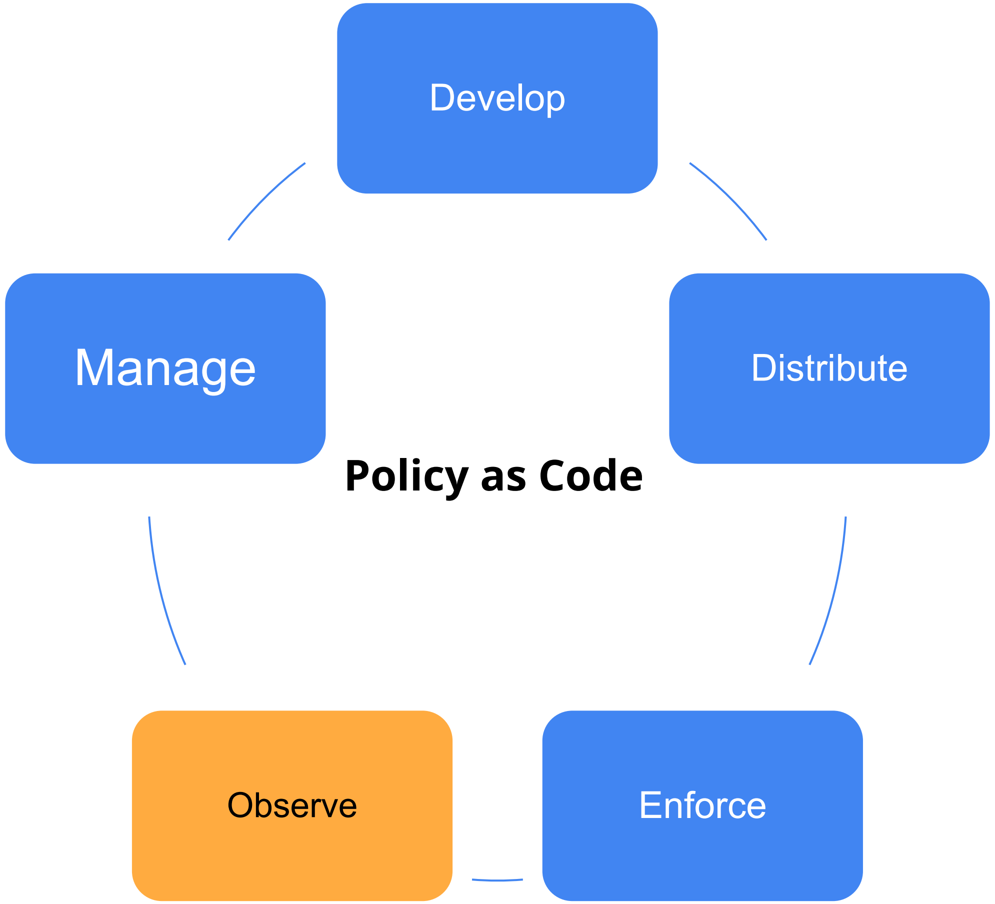
- Enforce in CI/CD pipelines
- Enforce at admission controls
- Enforce via background scans



Cloud Native Policy Management

Observe:

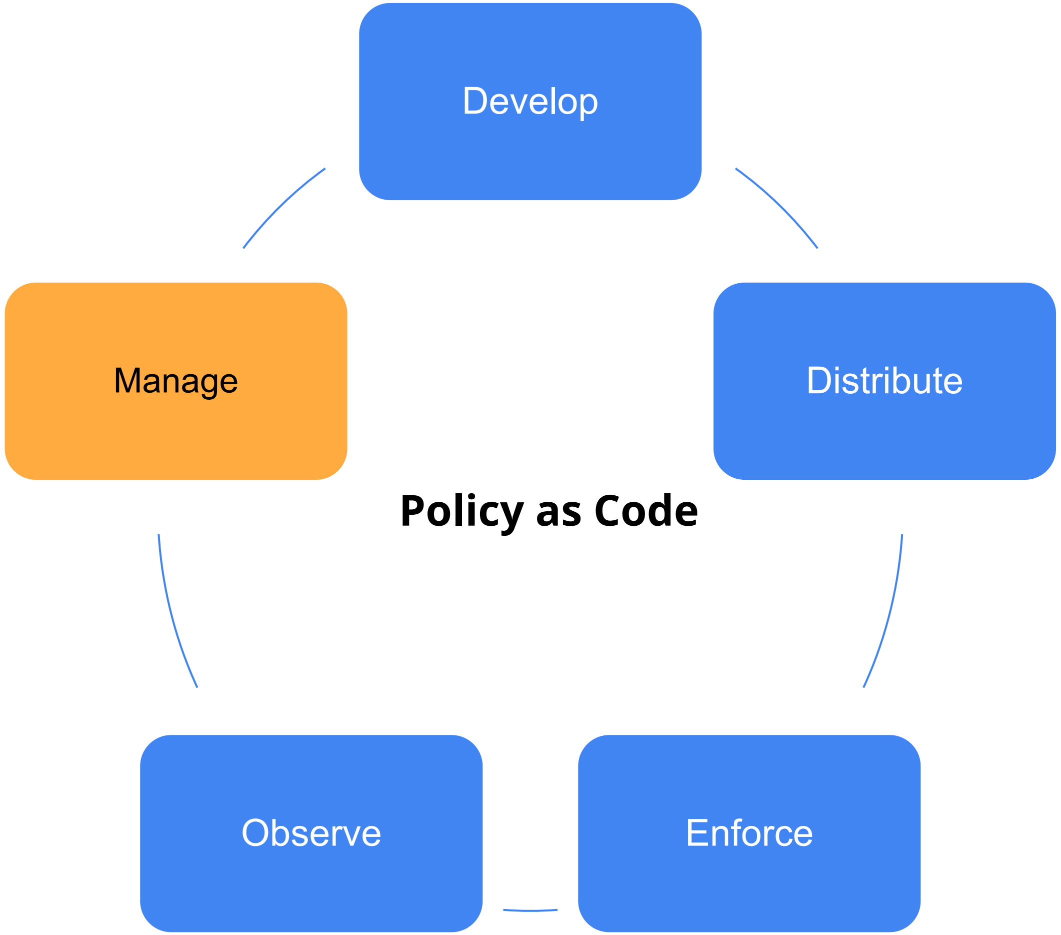
- Policy reporting via Kubernetes APIs
- Policy events
- Policy metrics
- Engine health and metrics



Cloud Native Policy Management

Manage:

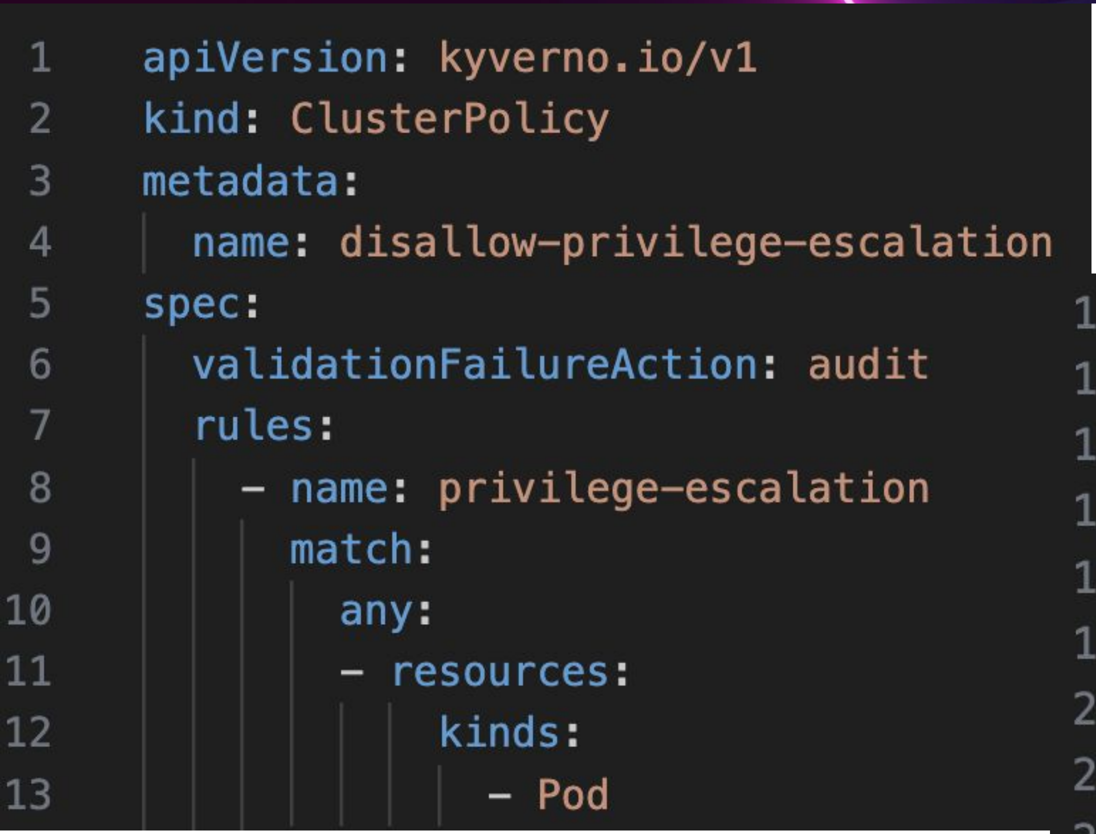
- Flexible rollout for policies
- Policy exceptions
- Remediation
- Scalable enforcement



Use case: Pod Security

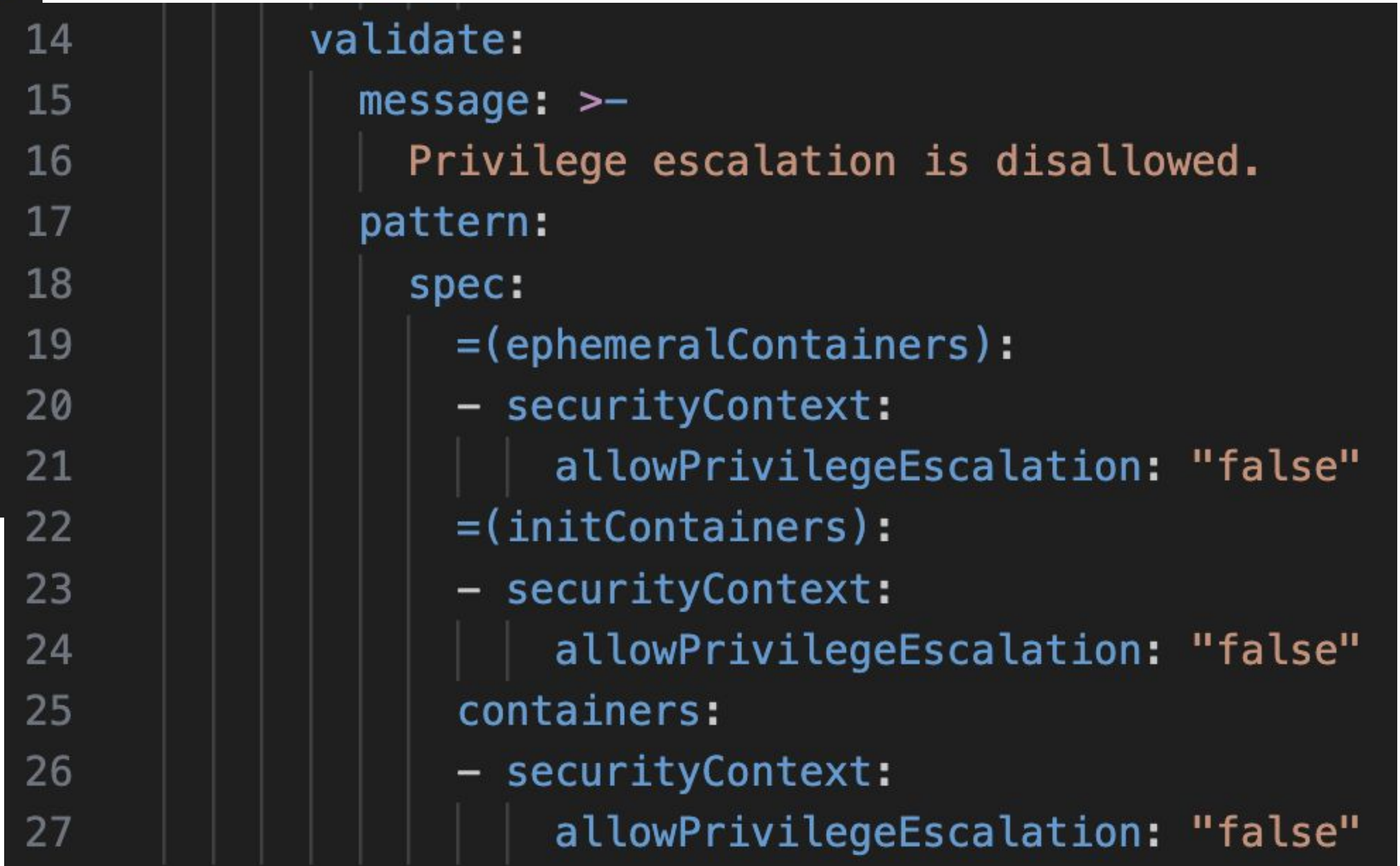
- Single cluster-wide policy
- Extends Pod Security Admission
- Enforce namespace labels
- Flexible exception management

Use case: Pod Security



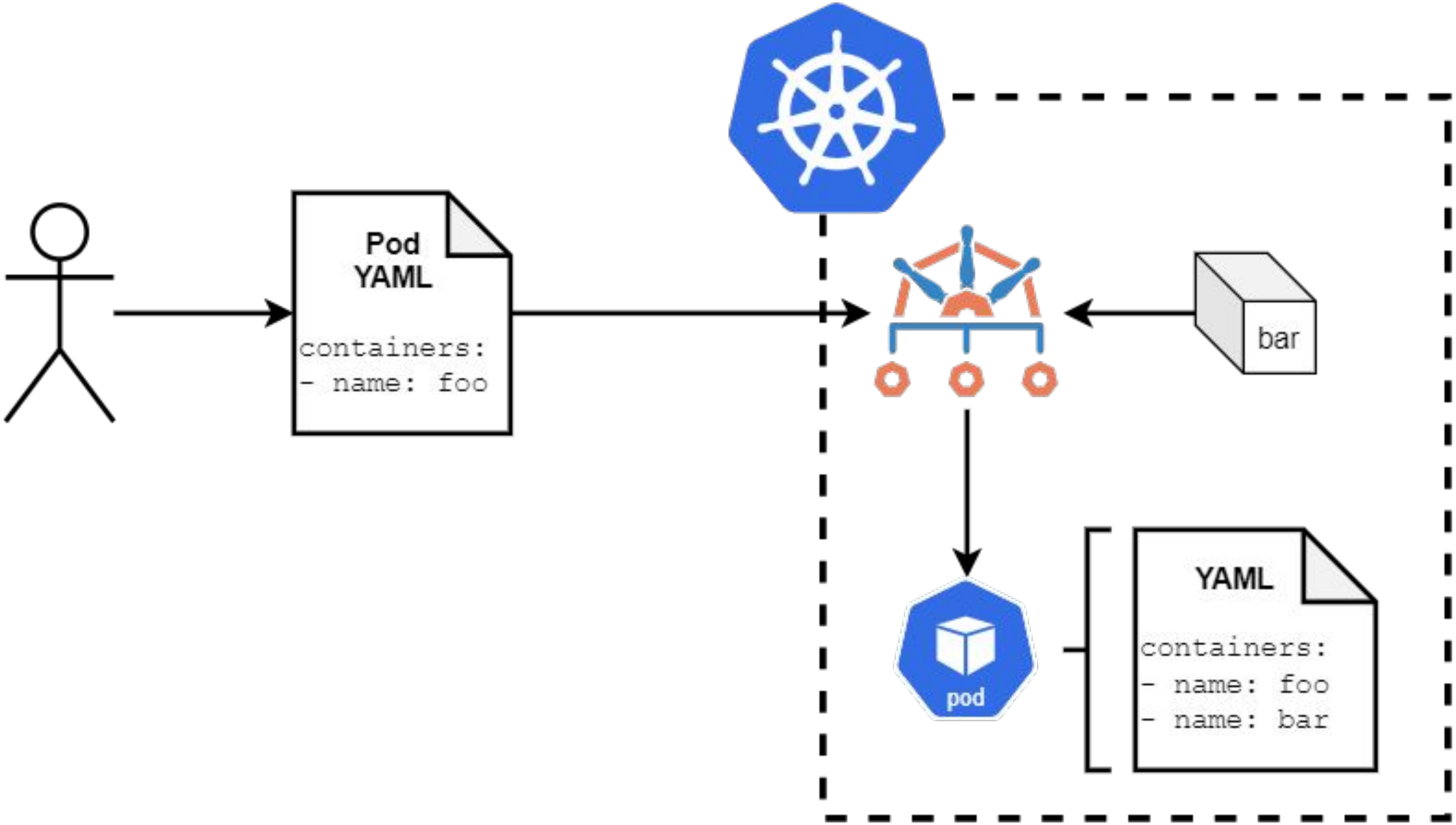
```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-privilege-escalation
spec:
  validationFailureAction: audit
  rules:
    - name: privilege-escalation
      match:
        any:
        - resources:
            kinds:
              - Pod
```

💥Disallow privilege escalation

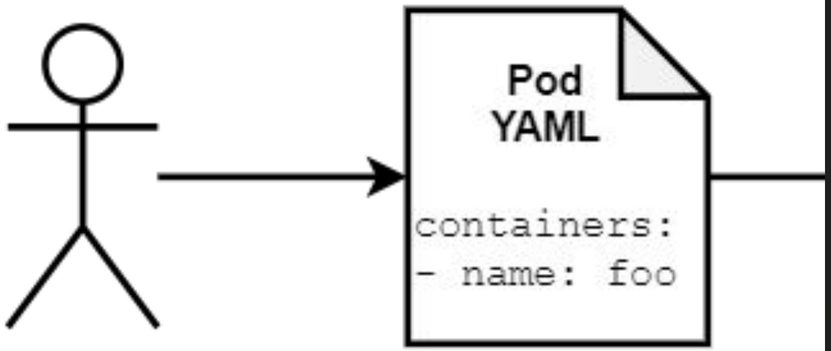


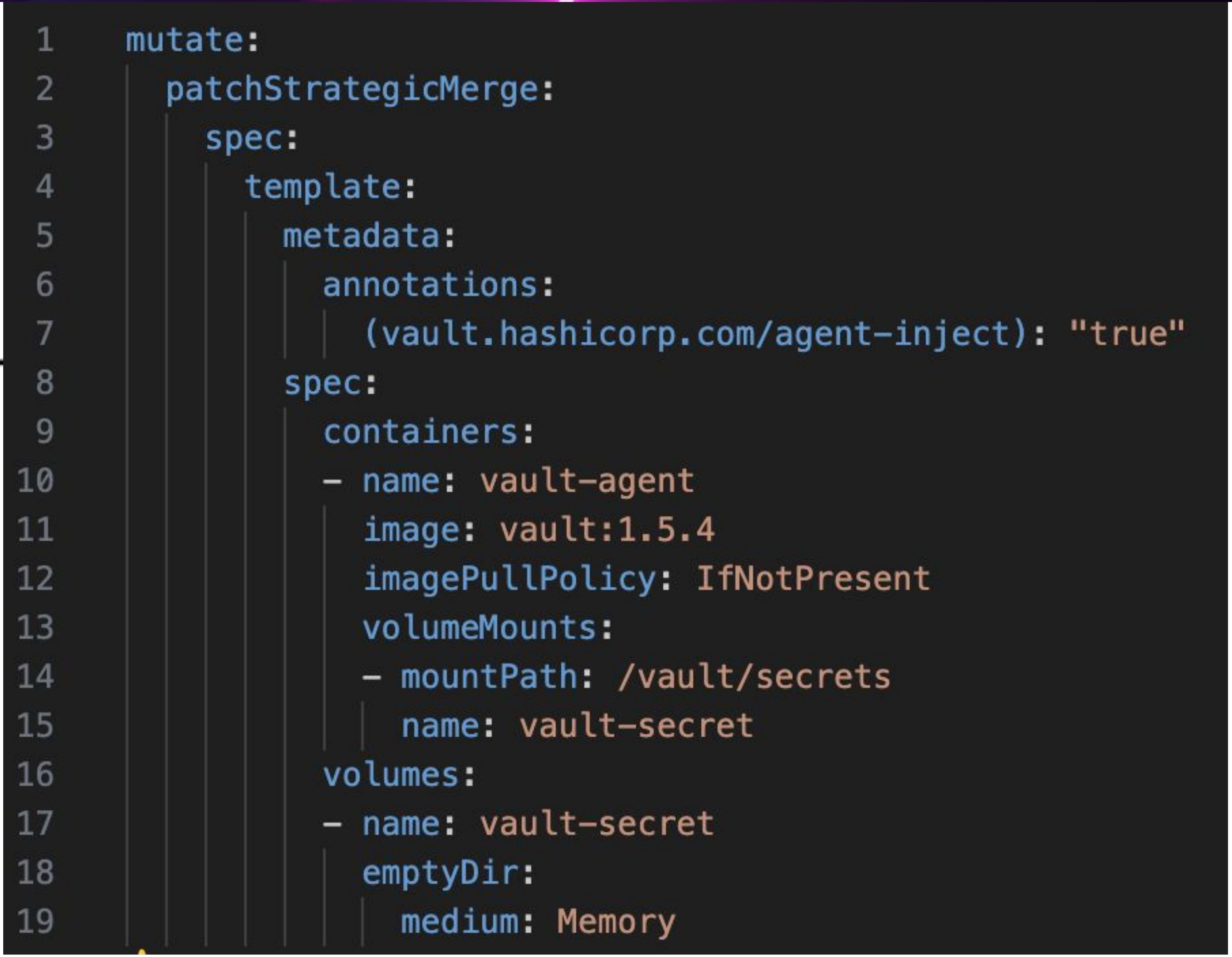
```yaml
      validate:
        message: >-
          Privilege escalation is disallowed.
        pattern:
          spec:
            =(ephemeralContainers):
            - securityContext:
                allowPrivilegeEscalation: "false"
            =(initContainers):
            - securityContext:
                allowPrivilegeEscalation: "false"
            containers:
            - securityContext:
                allowPrivilegeEscalation: "false"
```

Use case: Sidecar Injection



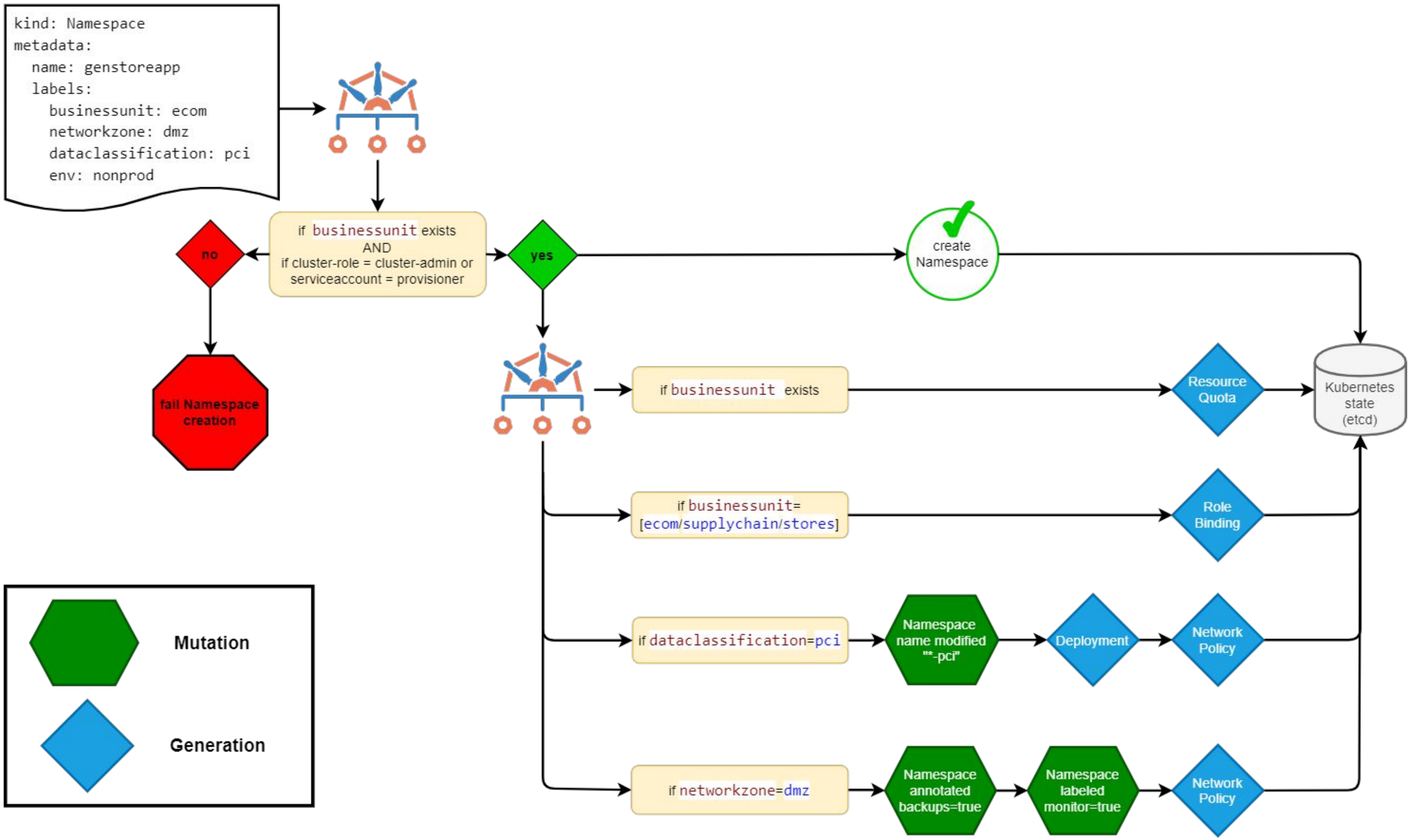
Use case: Sidecar Injection





```yaml
mutate:
  patchStrategicMerge:
    spec:
      template:
        metadata:
          annotations:
            (vault.hashicorp.com/agent-inject): "true"
        spec:
          containers:
          - name: vault-agent
            image: vault:1.5.4
            imagePullPolicy: IfNotPresent
            volumeMounts:
            - mountPath: /vault/secrets
              name: vault-secret
          volumes:
          - name: vault-secret
            emptyDir:
              medium: Memory
```

Use case: Multi-tenancy



```
kind: Namespace
metadata:
  name: genstoreapp
  labels:
    businessunit: ecom
    networkzone: dmz
    dataclassification: pci
    env: nonprod
```

if businessunit exists AND if cluster-role = cluster-admin or serviceaccount = provisioner

no

fail Namespace creation

yes

create Namespace

if businessunit exists

Resource Quota

Kubernetes state (etcd)

if businessunit= [ecom/supplychain/stores]

Role Binding

if dataclassification=pci

Namespace name modified "*-pci"

Deployment

Network Policy

if networkzone=dmz

Namespace annotated backups=true

Namespace labeled monitor=true

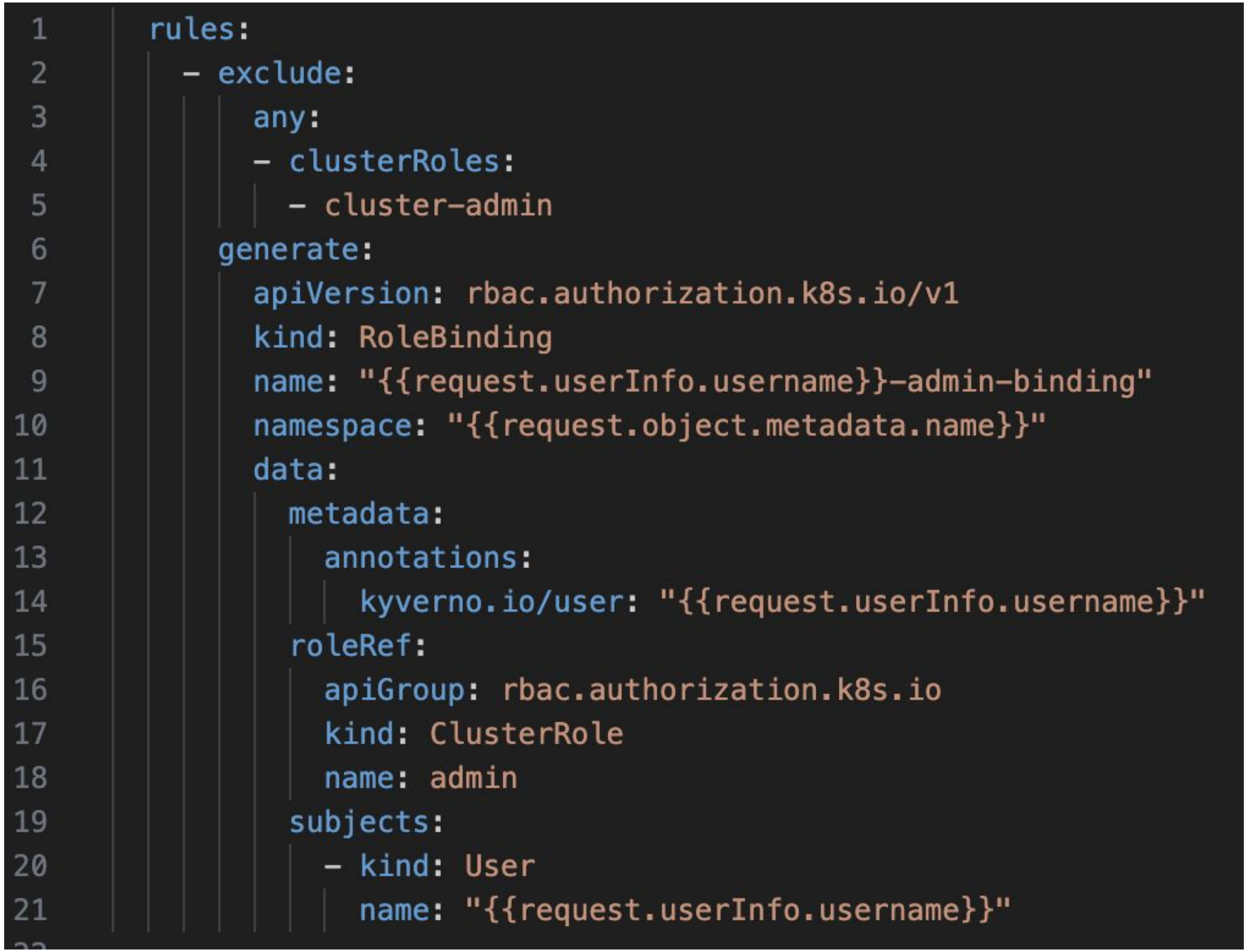
Network Policy

Mutation

Generation

Use case: Multi-tenancy

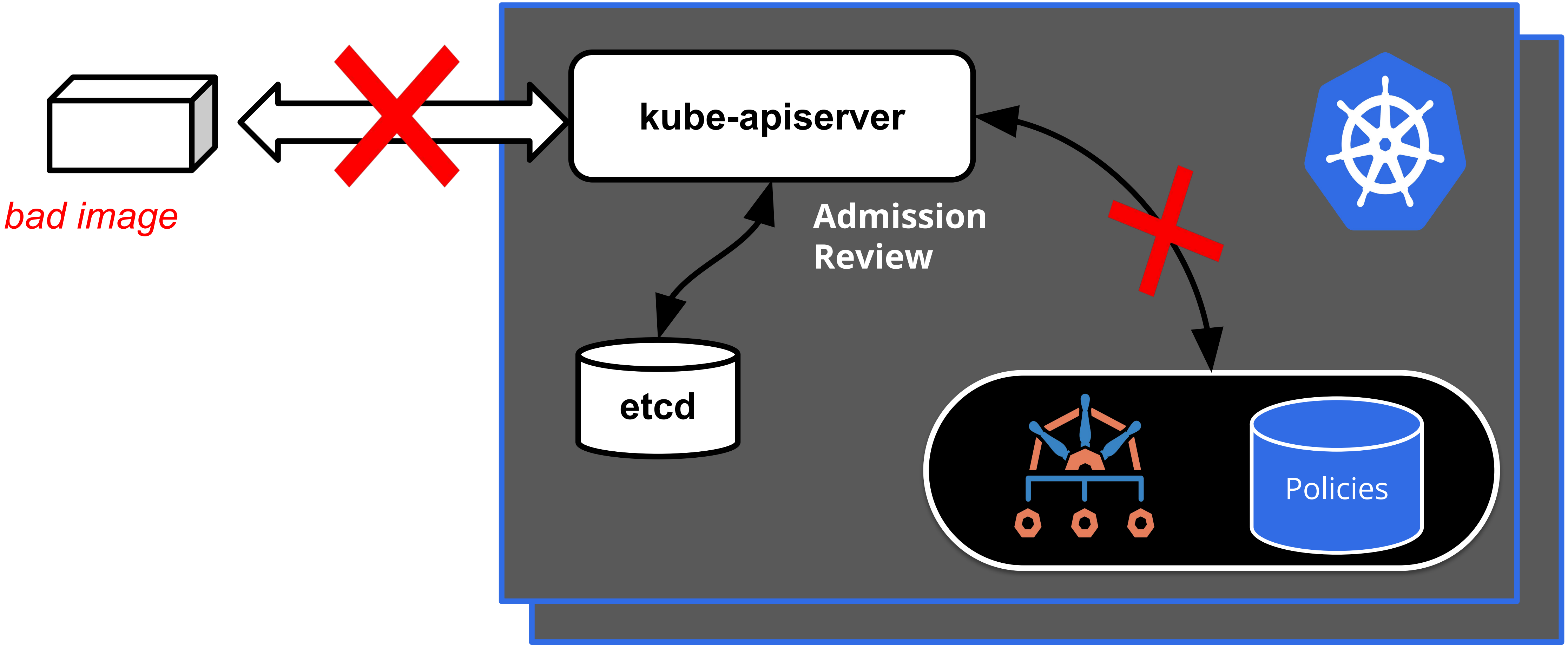
Auto-Generate Rolebinding



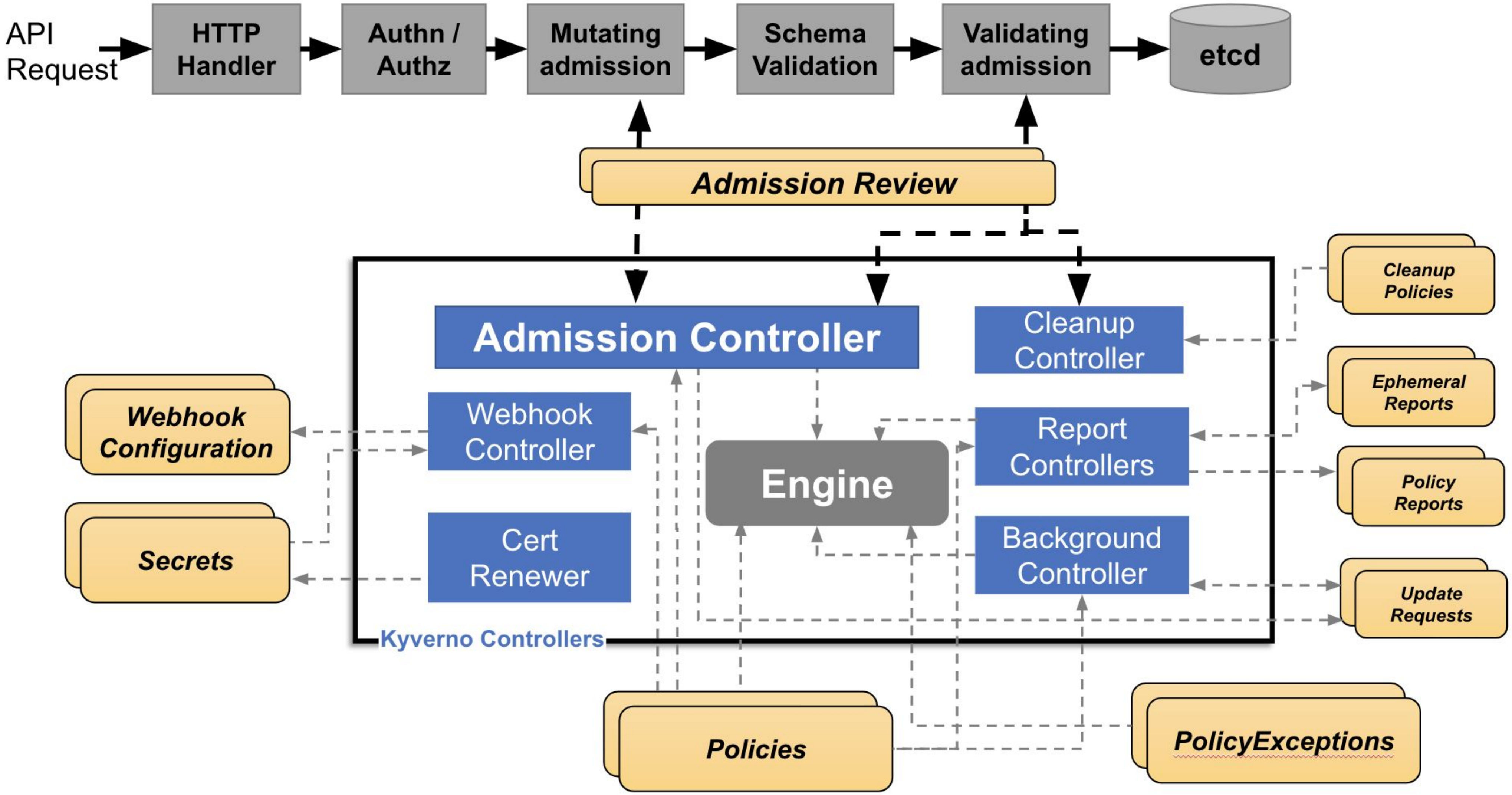
```yaml
rules:
  - exclude:
    any:
    - clusterRoles:
      - cluster-admin
  generate:
    apiVersion: rbac.authorization.k8s.io/v1
    kind: RoleBinding
    name: "{{request.userInfo.username}}-admin-binding"
    namespace: "{{request.object.metadata.name}}"
    data:
      metadata:
        annotations:
          kyverno.io/user: "{{request.userInfo.username}}"
      roleRef:
        apiGroup: rbac.authorization.k8s.io
        kind: ClusterRole
        name: admin
      subjects:
        - kind: User
          name: "{{request.userInfo.username}}"
```

Software Supply Chain Security

Block bad images!



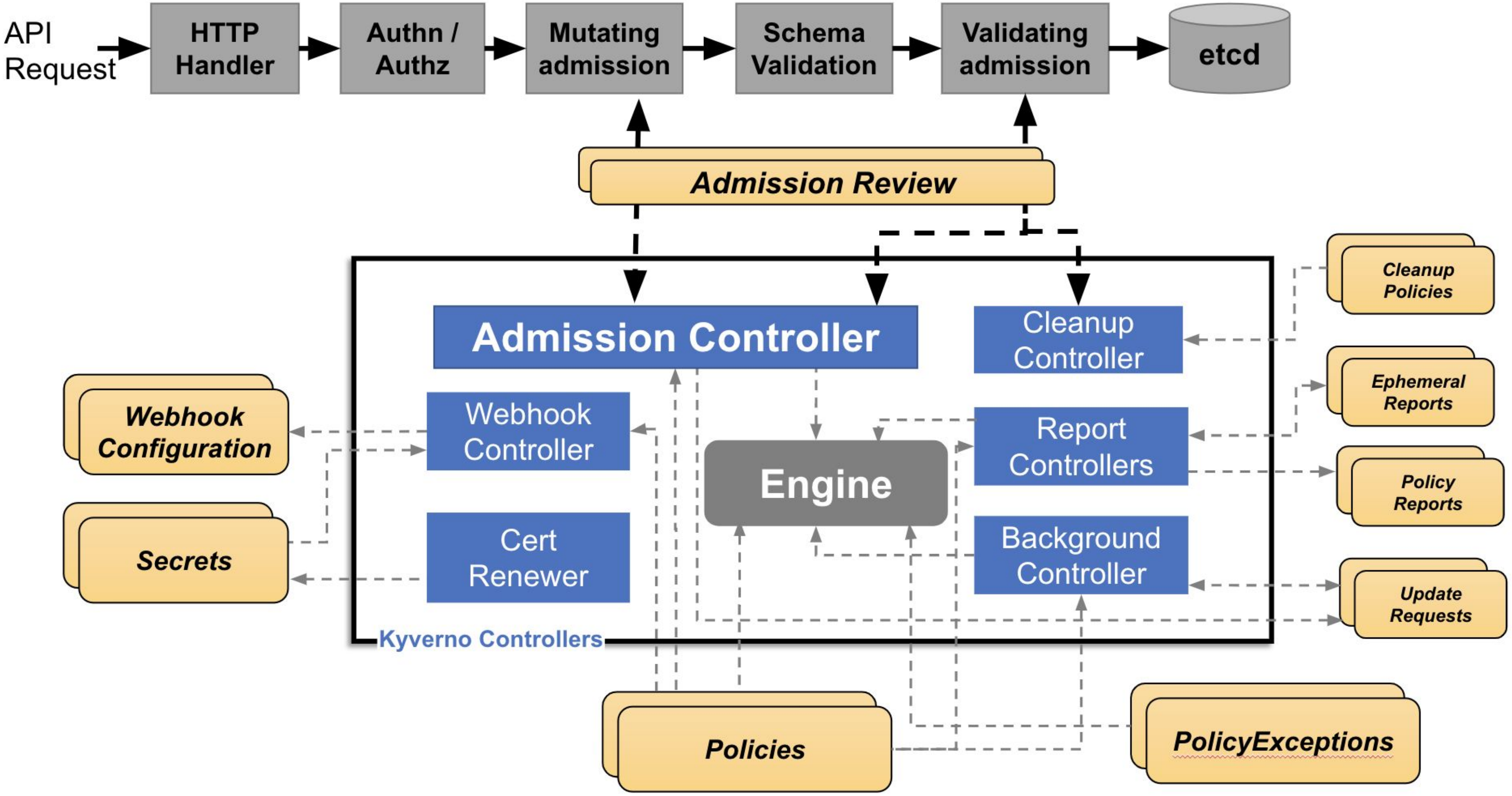
Kyverno Architecture



Deploy Kyverno from Scratch

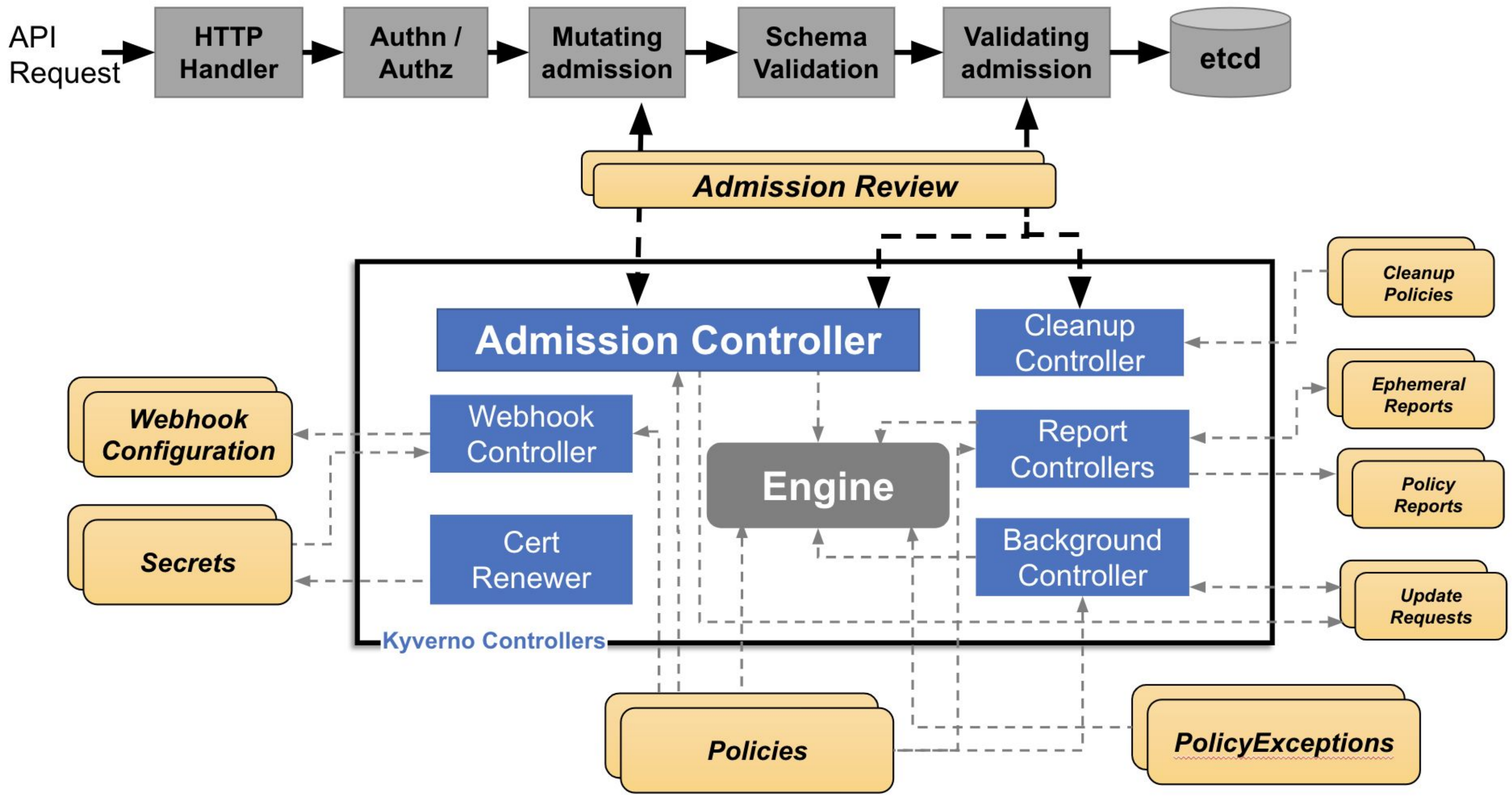
- Kyverno Playground
  - Kyverno Playground
- Kyverno CLI
  - Homebrew, Krew, GitHub Action, AUR, Binary
- Kyverno Controllers
  - Kubectl, Kustomize, GitOpts, Kubernetes APIs, etc

Admission Controller



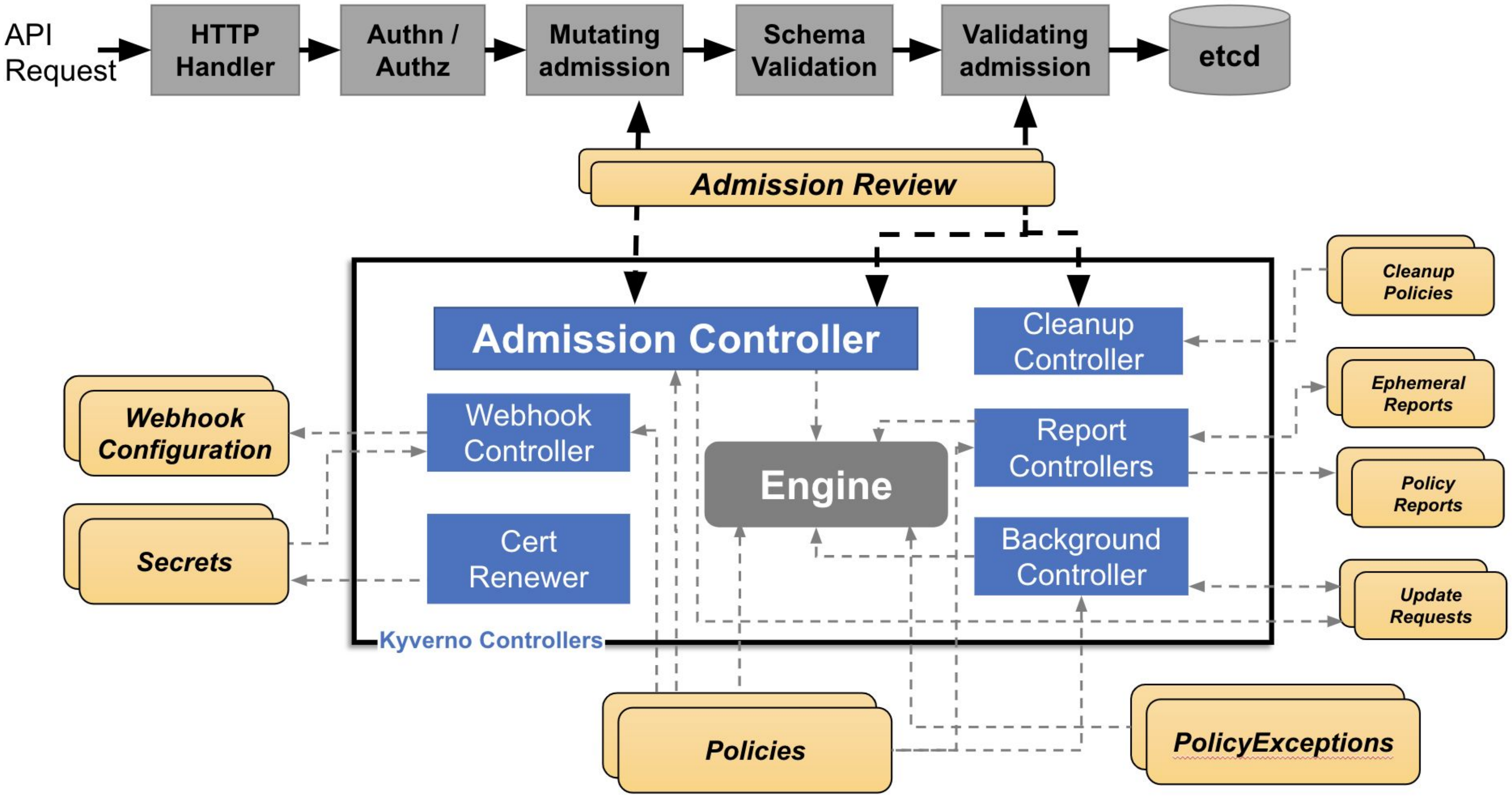
- Horizontal Scale
- Vertical Scale

Cleanup Controller



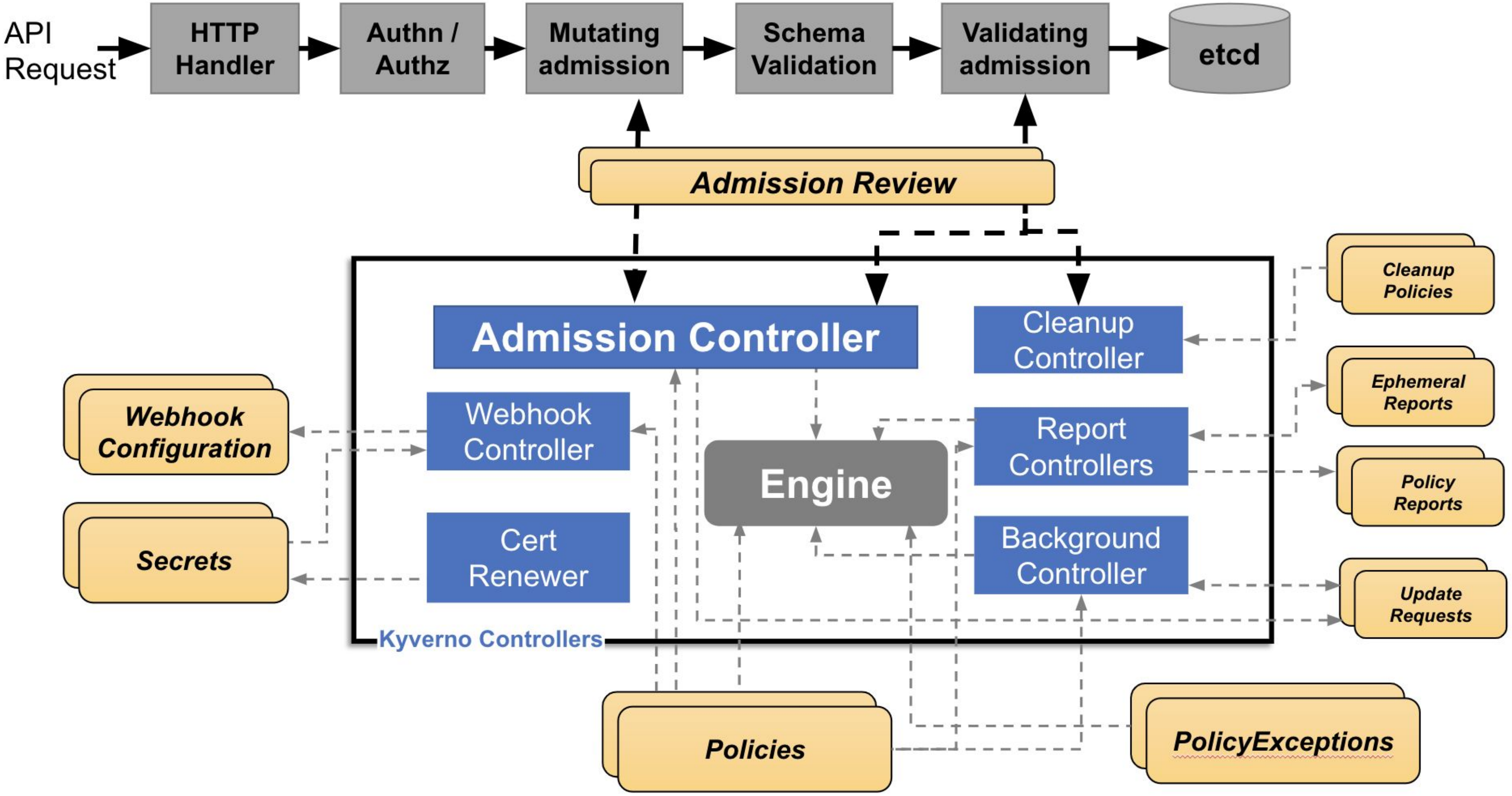
- Horizontal Scale
- Vertical Scale

Reports Controller



- Vertical Scale

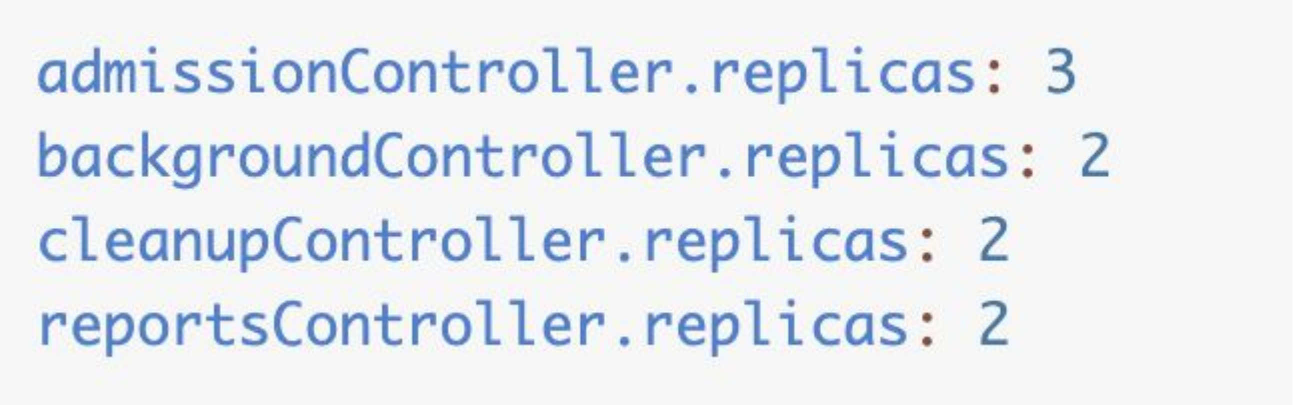
Background Controller



- Vertical Scale

Scaling Kyverno

- High Available Installation
  - Helm chart is the recommended
  - High availability is achieved on a per-controller basis
  - multiple replicas do not necessarily equate to higher scale or performance



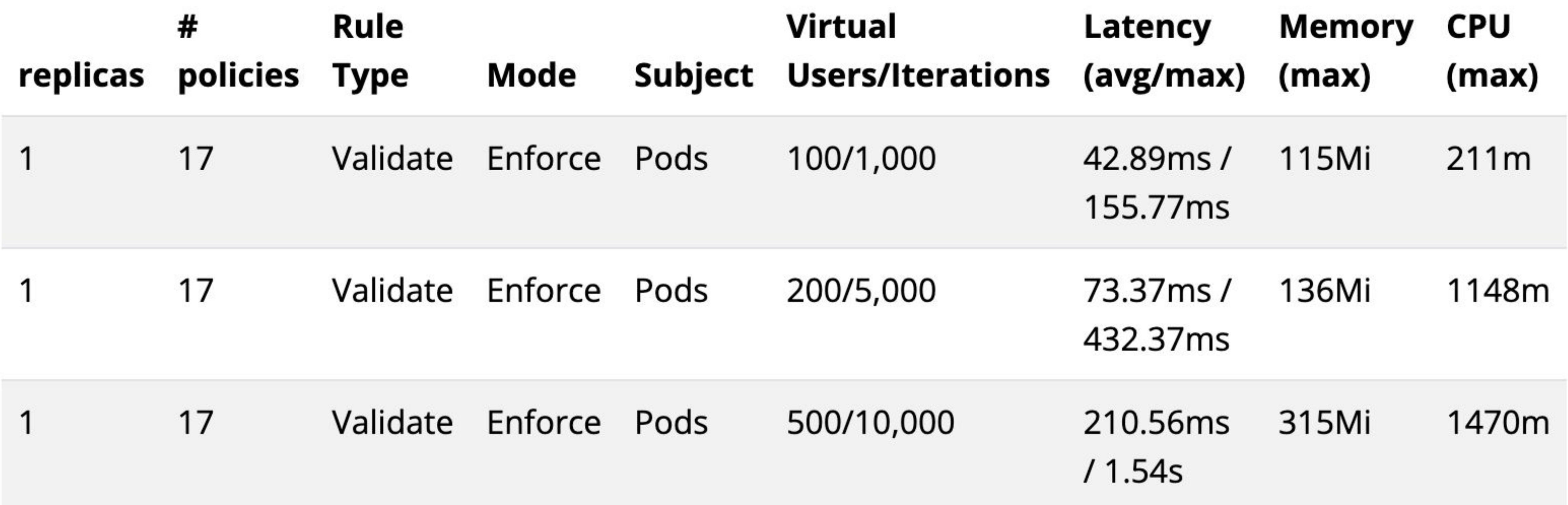
```
admissionController.replicas: 3
backgroundController.replicas: 2
cleanupController.replicas: 2
reportsController.replicas: 2
```

Tips on Scaling

- **Admission Review** - validate, mutate, image verification
  - Admission request rate
- **Reporting** - validate, image verification
  - In-memory cache for CRs ***policyreport, ephemeralreports***
  - Workers for reconciliation
- **Background policies** - generate, mutate existing
  - In-memory cache for CR ***updaterequest***
  - Workers for reconciliation
- **Others**
  - Adjust client side QPS & Burst for throttling issues
  - GlobalContextEntry for caching

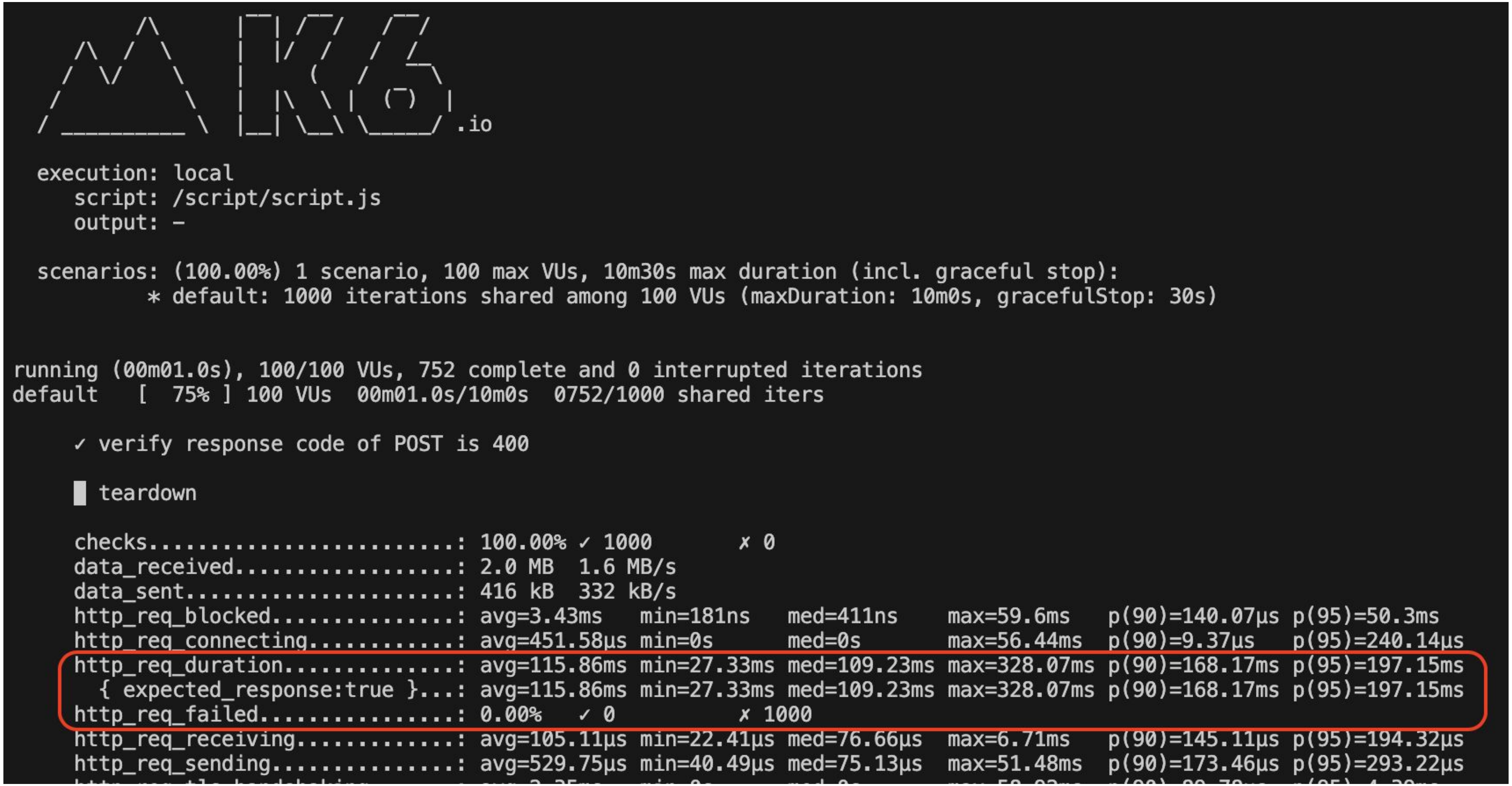
Performance

- Scaling testing data is published for each release
- Conduct tests using Kwok and K6
  - Kwok simulates clusters of thousands of Nodes
  - K6 is OSS for load testing



replicas	# policies	Rule Type	Mode	Subject	Virtual Users/Iterations	Latency (avg/max)	Memory (max)	CPU (max)
1	17	Validate	Enforce	Pods	100/1,000	42.89ms / 155.77ms	115Mi	211m
1	17	Validate	Enforce	Pods	200/5,000	73.37ms / 432.37ms	136Mi	1148m
1	17	Validate	Enforce	Pods	500/10,000	210.56ms / 1.54s	315Mi	1470m

Performance



```
          /\      |‾‾| /‾‾/   /‾‾/
     /\  /  \     |  |/  /   /  /
    /  \/    \    |     (   /   ‾‾\
   /          \   |  |\  \ |  (‾)  |
  / __________ \  |__| \__\ \_____/ .io

  execution: local
     script: /script/script.js
     output: -

  scenarios: (100.00%) 1 scenario, 100 max VUs, 10m30s max duration (incl. graceful stop):
           * default: 1000 iterations shared among 100 VUs (maxDuration: 10m0s, gracefulStop: 30s)


running (00m01.0s), 100/100 VUs, 752 complete and 0 interrupted iterations
default   [  75% ] 100 VUs  00m01.0s/10m0s  0752/1000 shared iters

     ✓ verify response code of POST is 400

     █ teardown

     checks.........................: 100.00% ✓ 1000       ✗ 0
     data_received..................: 2.0 MB  1.6 MB/s
     data_sent......................: 416 kB  332 kB/s
     http_req_blocked...............: avg=3.43ms   min=181ns   med=411ns    max=59.6ms   p(90)=140.07µs p(95)=50.3ms
     http_req_connecting............: avg=451.58µs min=0s      med=0s       max=56.44ms  p(90)=9.37µs   p(95)=240.14µs
     http_req_duration..............: avg=115.86ms min=27.33ms med=109.23ms max=328.07ms p(90)=168.17ms p(95)=197.15ms
       { expected_response:true }...: avg=115.86ms min=27.33ms med=109.23ms max=328.07ms p(90)=168.17ms p(95)=197.15ms
     http_req_failed................: 0.00%   ✓ 0          ✗ 1000
     http_req_receiving.............: avg=105.11µs min=22.41µs med=76.66µs  max=6.71ms   p(90)=145.11µs p(95)=194.32µs
     http_req_sending...............: avg=529.75µs min=40.49µs med=75.13µs  max=51.48ms  p(90)=173.46µs p(95)=293.22µs
```

Join us

- **Slack channels** under Kubernetes and CNCF workspaces
- **Public meetings** for maintainers, contributors and community users
- **GitHub issues**

💥Maintainers & Contributors💥