Kyverno Admission Controller

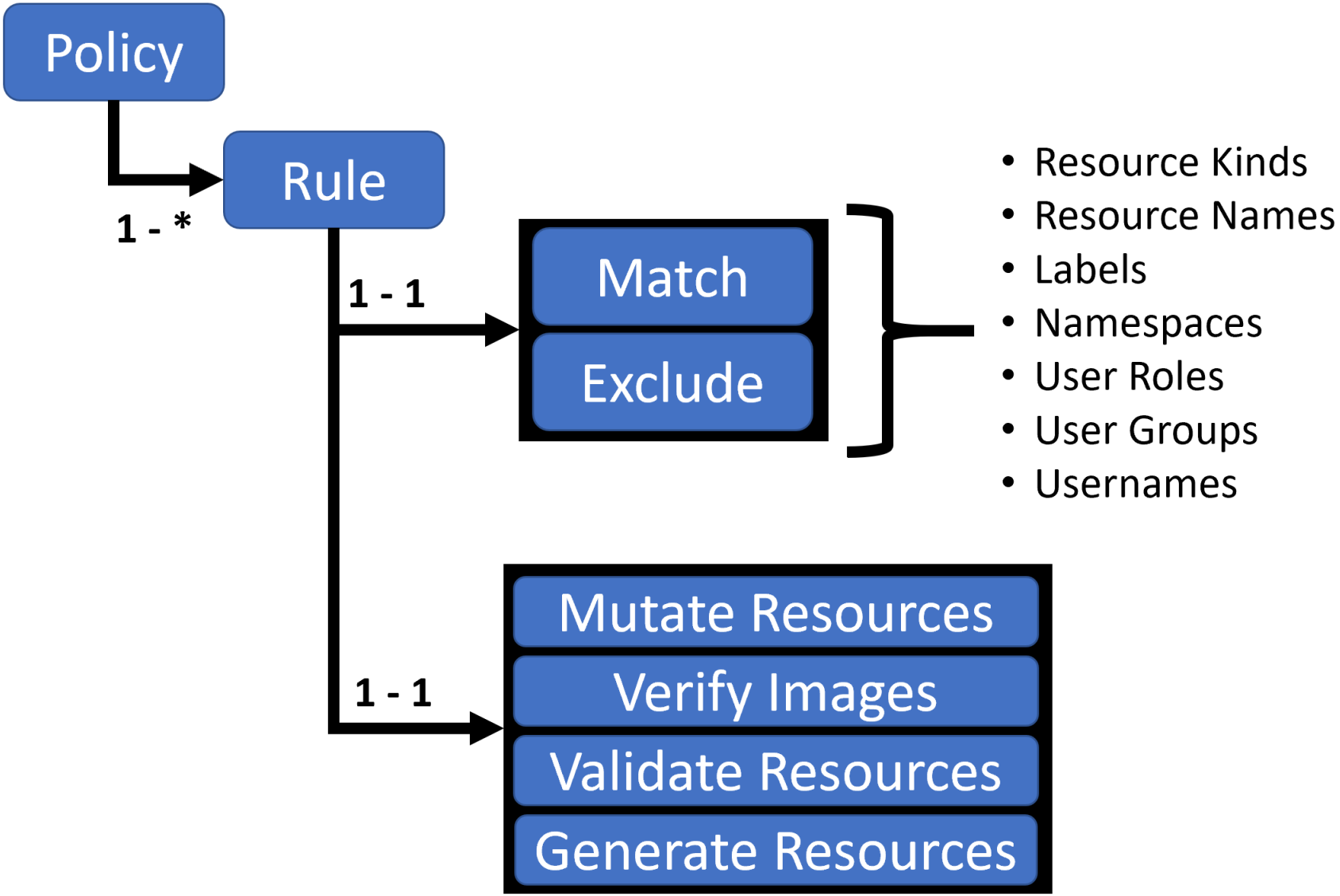
Kyverno Introduction

Kyverno is another admission controller, like Pod Security Policies, Pod Security Standards and OPA Gatekeeper.

Kyverno is roughly as powerful as OPA Gatekeeper, but it doesn't require learning a new language.

It works by allowing you to spell out which part of the object's manifest you want to check and what you want to check it for.

Kyverno also has a large collection of pre-written rules.


Policy
Rule
1 - *
1 - 1
Match
Exclude
• Resource Kinds
• Resource Names
• Labels
• Namespaces
• User Roles
• User Groups
• Usernames
1 - 1
Mutate Resources
Verify Images
Validate Resources
Generate Resources

Example Kyverno Rule

The "match" section specifies what resources (objects) this rule applies to. In this example, we check pods.

The "validate" section's "message" is displayed if a resource breaks the rule.

The "pattern" specifies what element we are checking and what has to be matched.

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-owner-label
spec:
  validationFailureAction: enforce
  rules:
  - name: check-for-owner-label
    match:
      any:
      - resources:
          kinds:
          - Pod
    validate:
      message: "label 'owner' required"
      pattern:
        metadata:
          labels:
            owner: "?*"
```

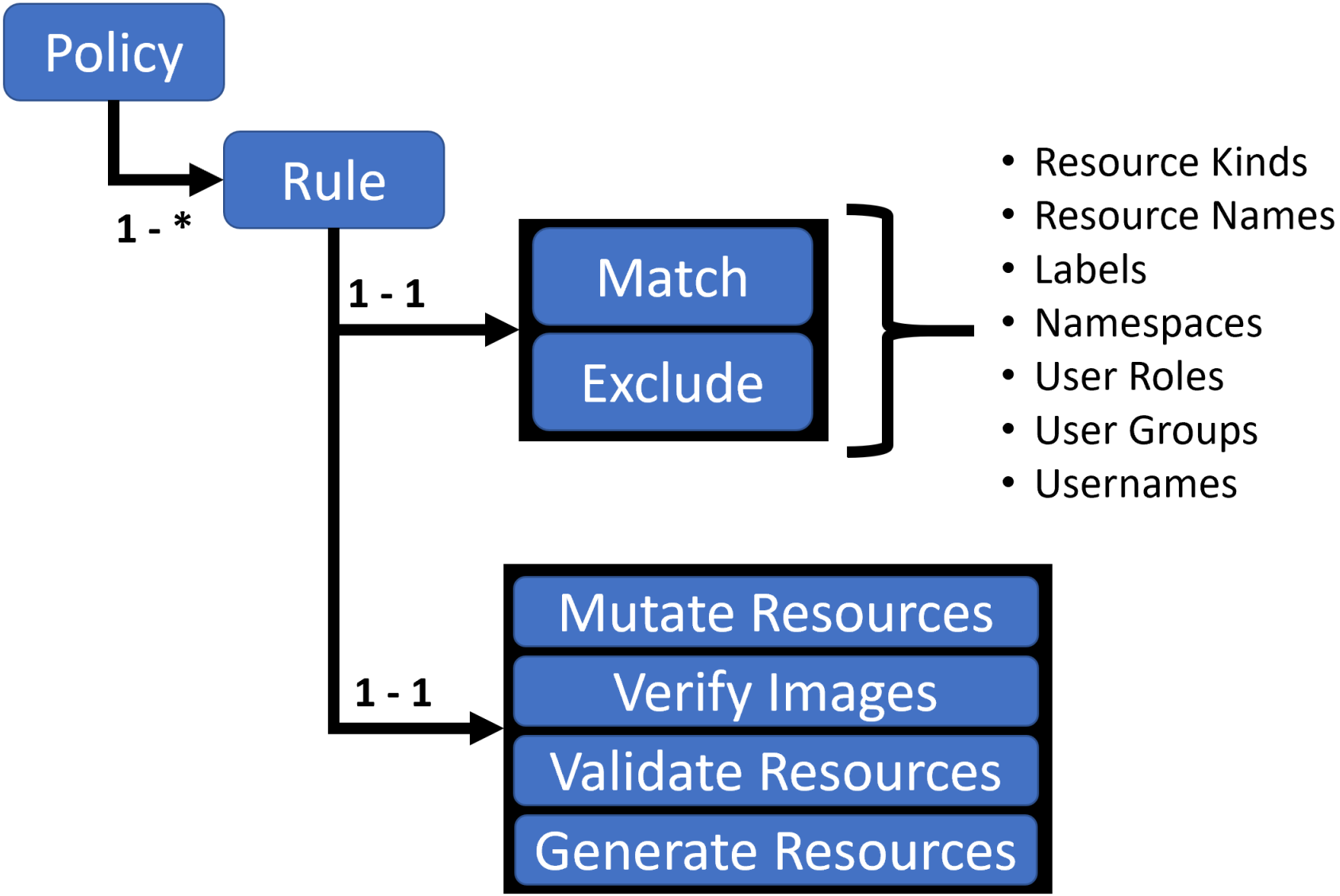
Validating and Mutating Admission Controllers

Unlike Pod Security Policies and Pod Security Standards, both Kyverno and OPA Gatekeeper can also modify (mutate) Kubernetes objects.

You tell Kyverno how to do this by specifying an RFC 6902 JSON Patch or a strategic merge patch.

In this example, we add a "serverip" data field to any configmap created in the "storage" namespace.

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: add-ip-to-server-configmap
spec:
  rules:
    - name: configmap-add-ip
      match:
        any:
        - resources:
            kinds:
              - ConfigMap
            namespace:
              - storage
  mutate:
        patchesJson6902: |-
          - path: "/data/serverip"
            op: add
            value: 169.254.169.254
```


Policy
Rule
1 - *
1 - 1
Match
Exclude
• Resource Kinds
• Resource Names
• Labels
• Namespaces
• User Roles
• User Groups
• Usernames
1 - 1
Mutate Resources
Verify Images
Validate Resources
Generate Resources

Kyverno Policy Library

Kyverno has a rich library of pre-written policies:

https://kyverno.io/policies/

Kyverno's policy library, like OPA Gatekeeper's, includes policies that match the functionality of Pod Security Policies.

Kyverno Example: Privileged Containers

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-privileged-containers
  annotations:
    policies.kyverno.io/title: Disallow Privileged Containers
    policies.kyverno.io/category: Pod Security Standards (Baseline)
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Pod
    kyverno.io/kyverno-version: 1.6.0
    kyverno.io/kubernetes-version: "1.22-1.23"
    policies.kyverno.io/description: >-
      Privileged mode disables most security mechanisms and must not be allowed. This policy
      ensures Pods do not call for privileged mode.
spec:
  validationFailureAction: audit
  background: true
  rules:
    - name: privileged-containers
      match:
        any:
        - resources:
            kinds:
              - Pod
      validate:
        message: >-
          Privileged mode is disallowed. The fields spec.containers[*].securityContext.privileged
          and spec.initContainers[*].securityContext.privileged must be unset or set to `false`.
        pattern:
          spec:
            =(ephemeralContainers):
              - =(securityContext):
                  =(privileged): "false"
            =(initContainers):
              - =(securityContext):
                  =(privileged): "false"
            containers:
              - =(securityContext):
                  =(privileged): "false"
```

Kyverno In Action

```
vagrant@bustakube-controlplane:~$ cat pod-ubuntu-priv.yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-priv
spec:
  containers:
  - image: ubuntu:22.04
    name: priv
    command:
    - /bin/sh
    - -c
    - sleep 360000
    securityContext:
      privileged: true
vagrant@bustakube-controlplane:~$ kubectl create -f pod-ubuntu-priv.yaml
pod/ubuntu-priv created
vagrant@bustakube-controlplane:~$
vagrant@bustakube-controlplane:~$ kubectl create -f disallow-privileged-containers.yaml
clusterpolicy.kyverno.io/disallow-privileged-containers created
vagrant@bustakube-controlplane:~$
vagrant@bustakube-controlplane:~$ kubectl create -f pod-ubuntu-priv-2.yaml
Error from server: error when creating "pod-ubuntu-priv-2.yaml": admission webhook "validate.kyverno.svc-fail" denied the request:

resource Pod/default/ubuntu-priv-2 was blocked due to the following policies

disallow-privileged-containers:
  privileged-containers: 'validation error: Privileged mode is disallowed. The fields
    spec.containers[*].securityContext.privileged and spec.initContainers[*].securityContext.privileged
    must be unset or set to `false`. Rule privileged-containers failed at path /spec/containers/0/securityContext/privileged/'
vagrant@bustakube-controlplane:~$
```