

# *Unmasked: Policing your Deployments with Open Policy Agents*

Policy as Code



 <https://linkedin.com/in/mjbright>

 @mjbright

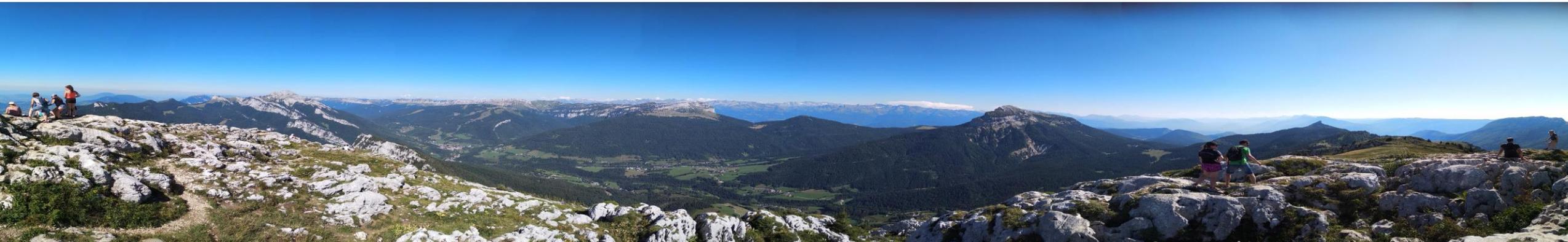
 @mjbright





# \$ who am i

- Trainer/consultant based in France
- Docker, Kubernetes, Terraform, Security
- Meetup Organizer



Grab a seat and let's get started ...



# Policy as Code / Policy Engines

Open Policy Agent & eco-system

Specialized tools: Kyverno

Summary



# Why Policy Engines?

# Why ?

We need ways of automatically checking policy adherence to secure the platform from attacks

Policy engines allow to apply a set of rules, which can **evolve** with security requirements **independently from the platform**

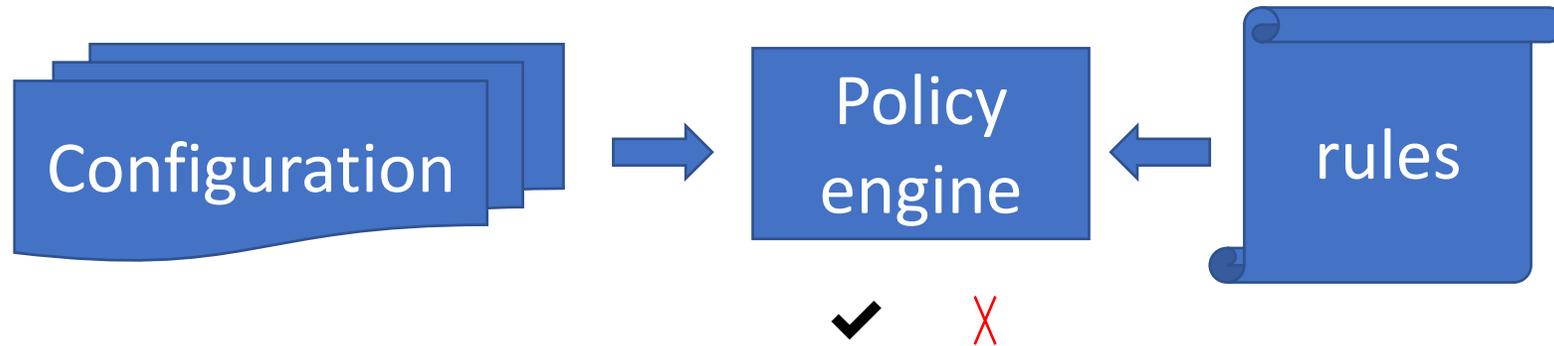
Policy engines may be

- open source or not
- declarative or not
- platform-specific or not

# Usage

# Usage (static)

## Static validation

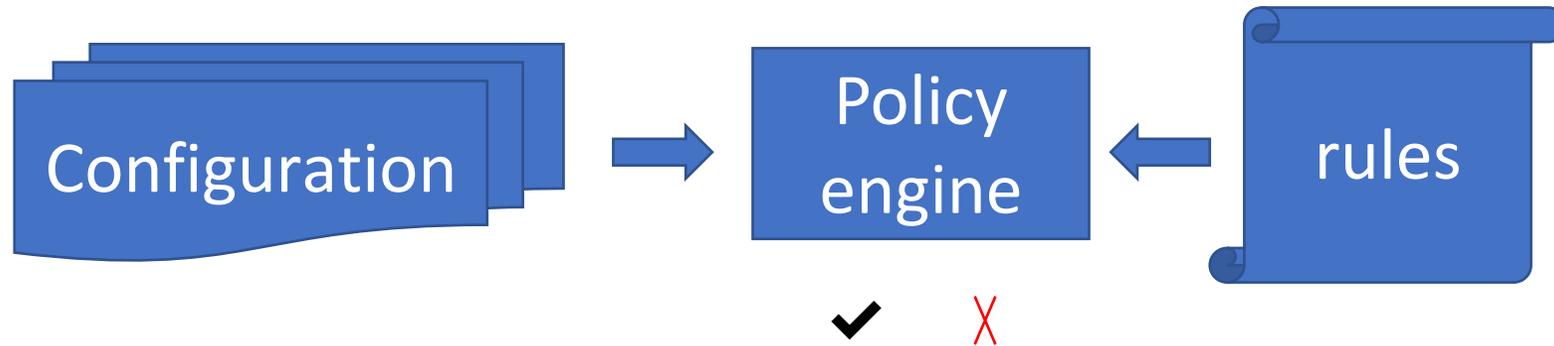


Can be used to validate

- Kubernetes YAML or JSON manifests
- Terraform configurations
- Many other types of configuration ... via conftest

# Usage (static)

## Static validation



Can be used to validate

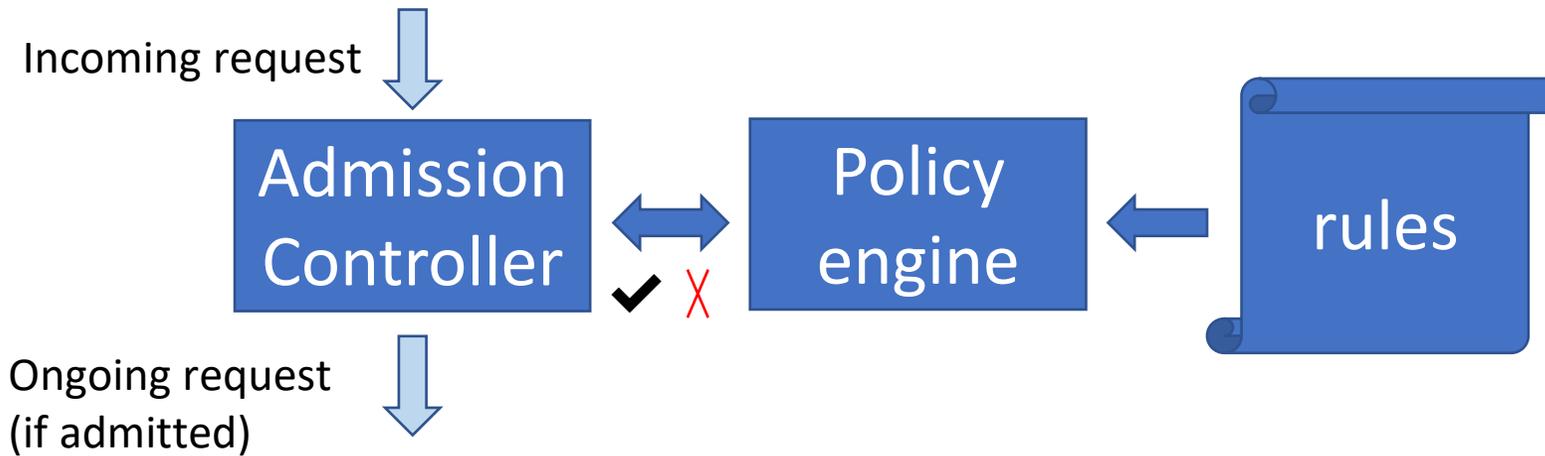
- Kubernetes YAML or JSON manifests
- Terraform configurations

e.g. as part of a CI/CD pipeline “shift left”



# Usage (dynamic)

Dynamic checks, e.g. intercepting API requests



Can be used to validate

- [enforce] allow/deny API requests
- [audit] log API requests

Can be used to mutate

- Modify the incoming request, to adhere to Policy

# Kubernetes Policy Enforcement

Kubernetes – “*pathologically extensible*”

# Kubernetes Policy Enforcement

Kubernetes – *“pathologically extensible”*

So we need to be able to adapt to that

*The good news is we can plugin a policy agent and update it's rules independently of Kubernetes*

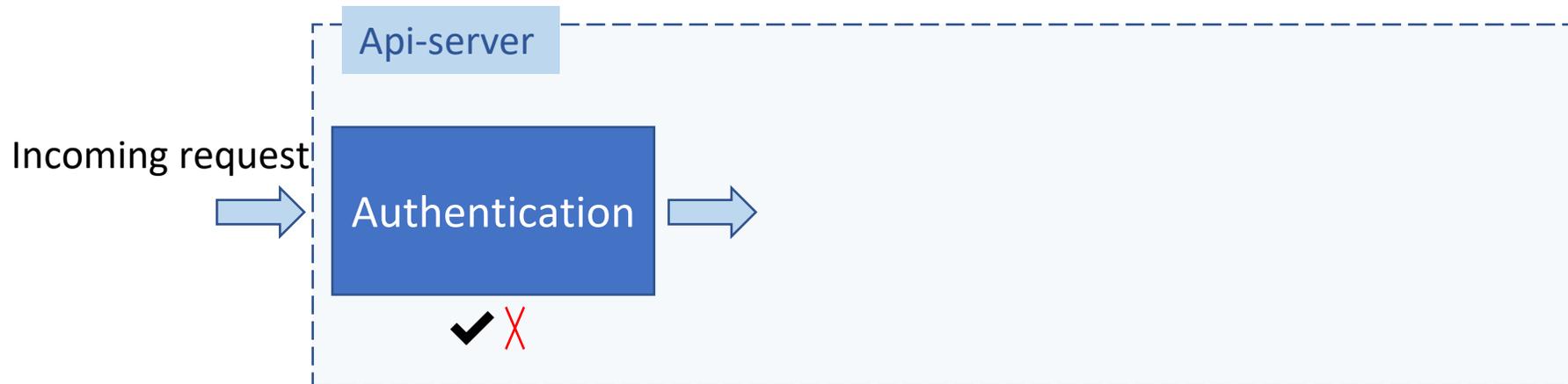
# Kubernetes Access Control

3 stages of Kubernetes (API) Access control

# Kubernetes Access Control

## 3 stages of Kubernetes (API) Access control

- Authentication: verify identity
- Authorisation: verify permissions
- Admission Control: verify request is allowed/valid

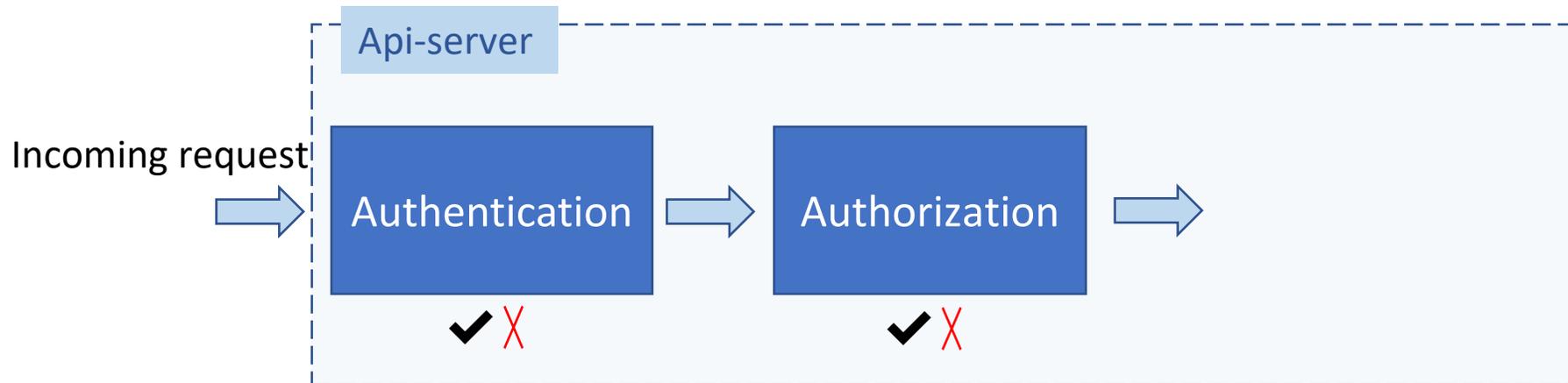


API requests from inside (Pods) or outside the cluster (kubectl, dashboard, ...)

# Kubernetes Access Control

## 3 stages of Kubernetes (API) Access control

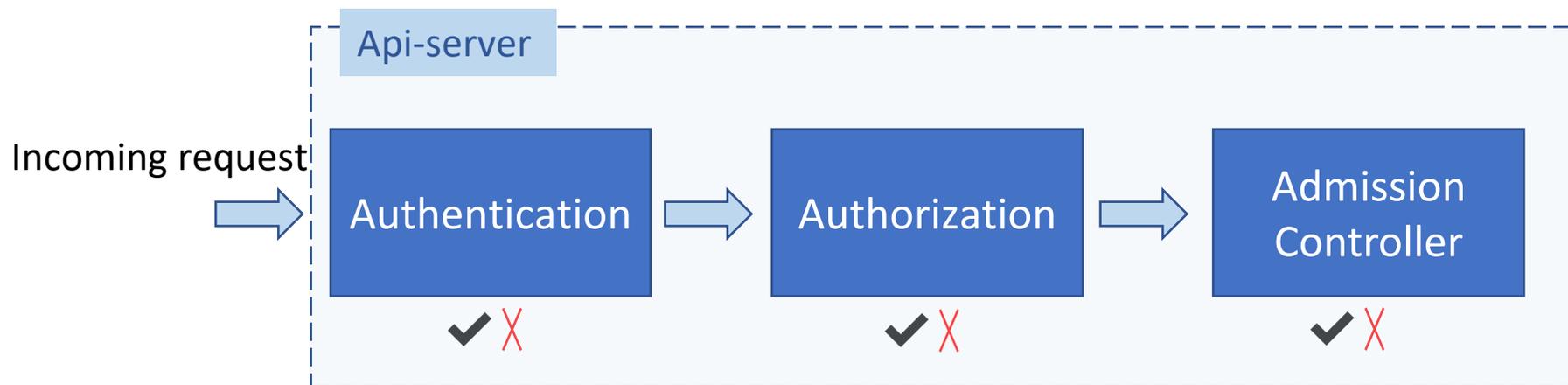
- Authentication: verify identity
- Authorisation: verify permissions
- Admission Control: verify request is allowed/valid



# Kubernetes Access Control

## 3 stages of Kubernetes (API) Access control

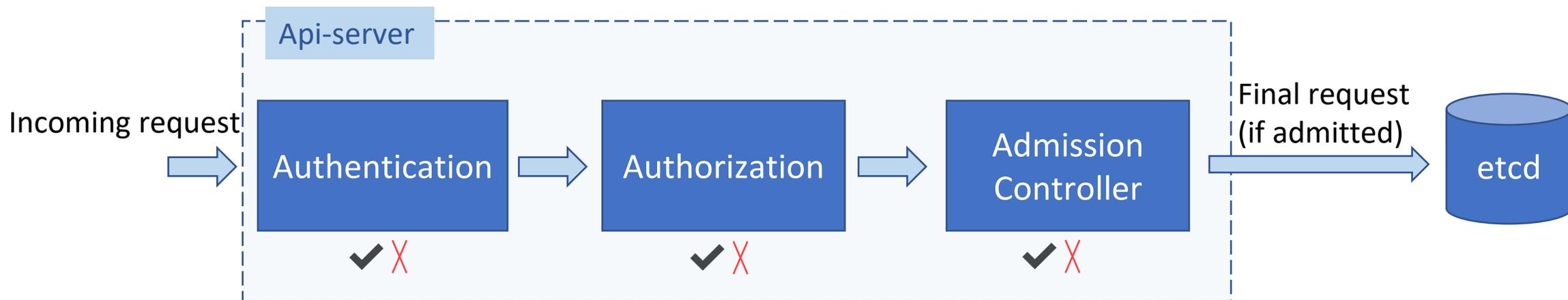
- Authentication: verify identity
- Authorisation: verify permissions
- Admission Control: verify request is allowed/valid



# Kubernetes Access Control

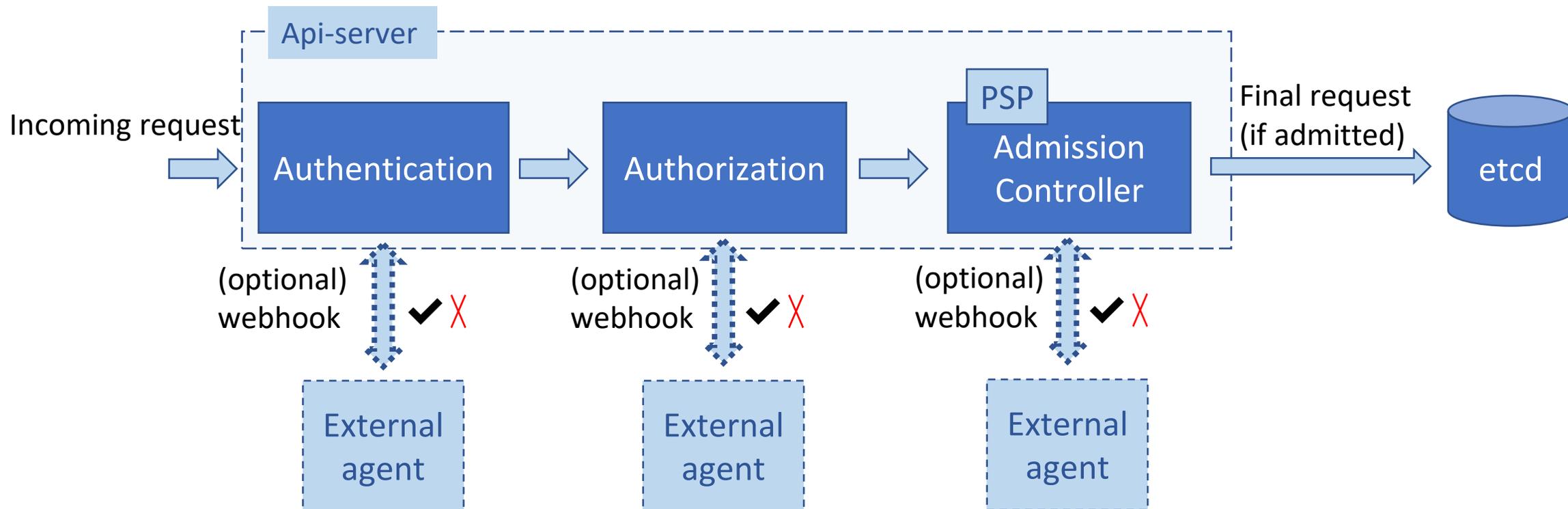
## 3 stages of Kubernetes (API) Access control

- Authentication: verify identity
- Authorisation: verify permissions
- Admission Control: verify request is allowed/valid



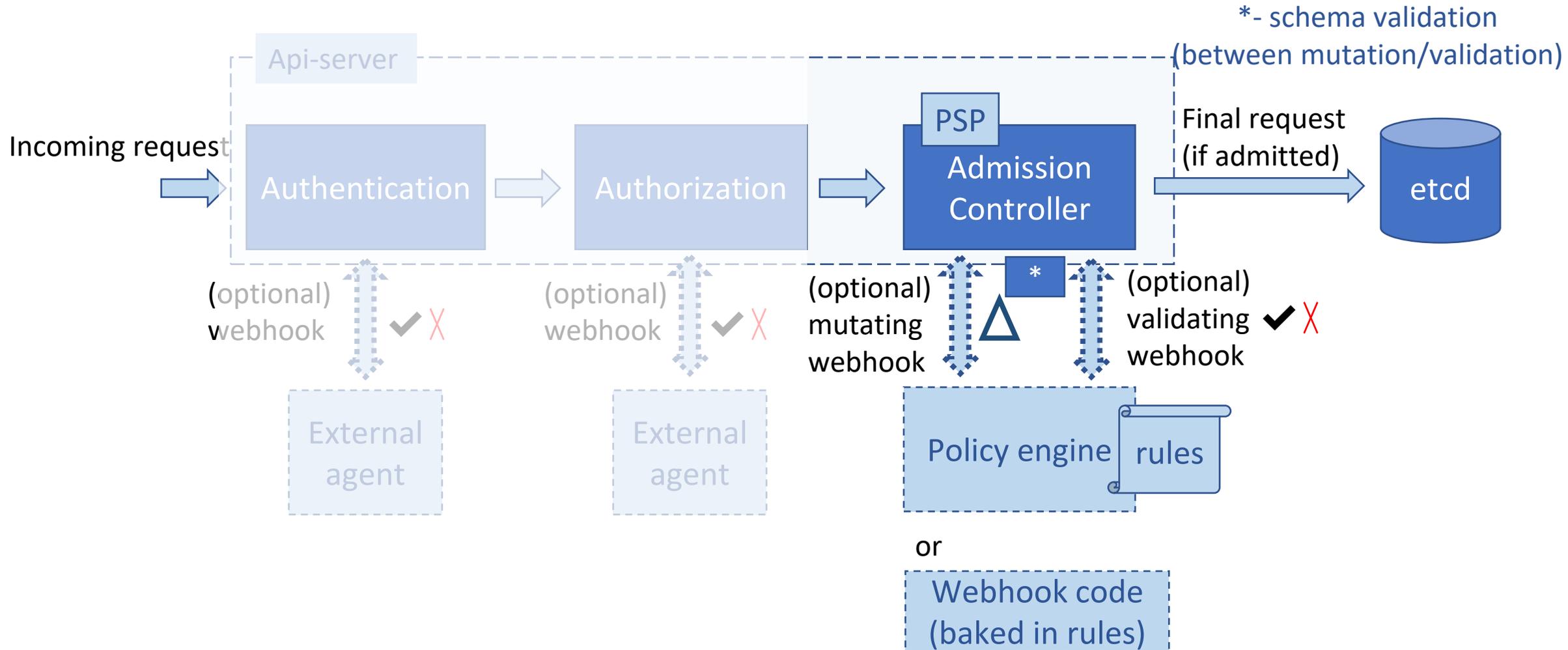
# Kubernetes Access Control

## Optional webhooks



# Kubernetes Access Control

## Optional webhooks



# Example Policies

# Example Kubernetes Policies

Validating webhook to enforce or log policy exceptions  
Mutating webhook to coerce request to adhere to policy

[psp:capabilities] Implementation of Pod Security Policies

[manageability] Enforce use of labels on Kubernetes resources

[manage costs] Prevent use of LoadBalancer service

[best practice] Require specification of a non-default namespace

[best practice] Enforce Requests/Limit specifications on PodSpec

[best practice] Enforce ResourceQuota/LimitRange on Namespaces

Policy as Code / Policy Engines

Open Policy Agent & eco-system

Specialized tools: Kyverno

Summary





# Open Policy Agent (OPA)

Generalized Policy Agent



# Open Policy Agent (OPA)

## Generalized Policy Agent

Open source project created by Styra

Generalized policy engine - not Kubernetes-specific - can be applied to many components of your diverse IT stack



Policies defined in the Rego language

Styra provide their DAS product as well as many open source tools  
3<sup>rd</sup>-party products and open source tools



# Open Policy Agent (OPA)

## OPA Open Source & free tools

Open Source OPA Policy Engine usable as REST API server, cli tool or API

Or via free tools

- rego playground <https://play.openpolicyagent.org/>
- VSCode extension <https://github.com/open-policy-agent/vscode-opa>

3<sup>rd</sup>-party tools

- Conftest <https://www.conftest.dev/>
- Fregot [rego toolkit]
- regula[rego lib for Terraform] <https://github.com/fugue/regula>



# Open Policy Agent (OPA)

## Rego Playground: edit, test, share rules and input/output

The Rego Playground Examples ▾  Coverage  Evaluate  Publish

```
1 package play
2
3 # Welcome to the Rego playground! Rego (pronounced "ray-go") is OPA's po
4 #
5 # Try it out:
6 #
7 # 1. Click Evaluate. Note: 'hello' is 'true'
8 # 2. Change "world" to "hello" in the INPUT panel. Click Evaluate. Not
9 # 3. Change "world" to "hello" on line 25 in the editor. Click Evaluat
10 #
11 # Features:
12 #
13 #     Examples  browse a collection of example policies
14 #     Coverage  view the policy statements that were executed
15 #     Evaluate  execute the policy with INPUT and DATA
16 #     Publish   share your playground and experiment with local depl
17 #     INPUT     edit the JSON value your policy sees under the 'input
18 # (resize) DATA edit the JSON value your policy sees under the 'data
19 #     OUTPUT    view the result of policy execution
20
21 default hello = false
22
23 hello {
24     m := input.message
25     m == "world"
26 }
```

**INPUT**

```
1 {
2     "message": "world"
3 }
```

**DATA**

**OUTPUT**

```
1
```



# Open Policy Agent (OPA)

## VSCode Extension

```
simple.rego
1 package http.authz
2
3 default allow = false
4
5 allow {
6   input.method = "GET"
7   input.path = ["salary", user]
8   input.user = user
9 }

input.json
1 {
2   "user": "bob",
3   "method": "GET",
4   "path": [
5     "salary",
6     "alice"
7   ]
8 }
```

```
results.json
1 // Found 1 result. Took 17ms.
2 [
3   {
4     "user": "alice"
5   }
6 ]
```

- Check Syntax on Save
- Reformat File on Save
- Evaluate Packages
- Evaluate Selections
- Partially Evaluate Selections
- Trace Selections
- Profile Selections
- Run Tests in Workspace
- Toggle Coverage in Workspace
- Toggle Coverage of Selections



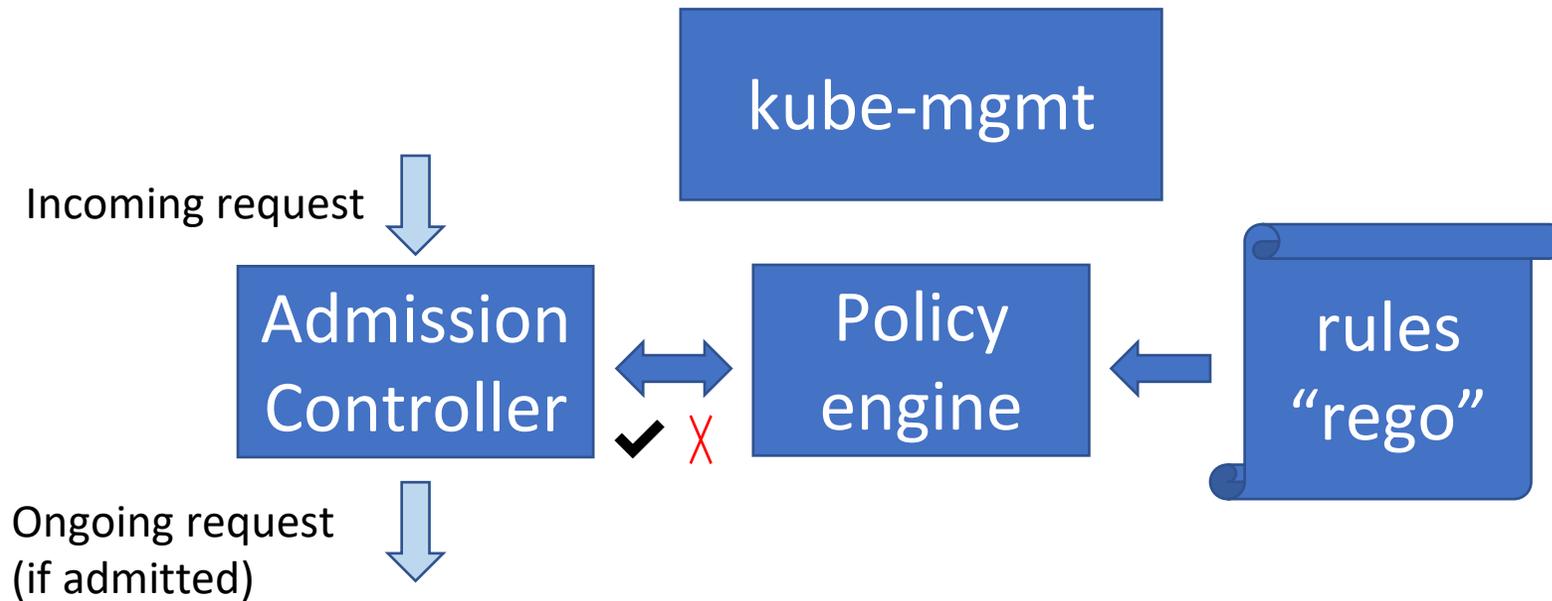
# Open Policy Agent (OPA)

## OPA & Kubernetes



# Open Policy Agent (OPA)

## OPA & Kubernetes (GK v1)

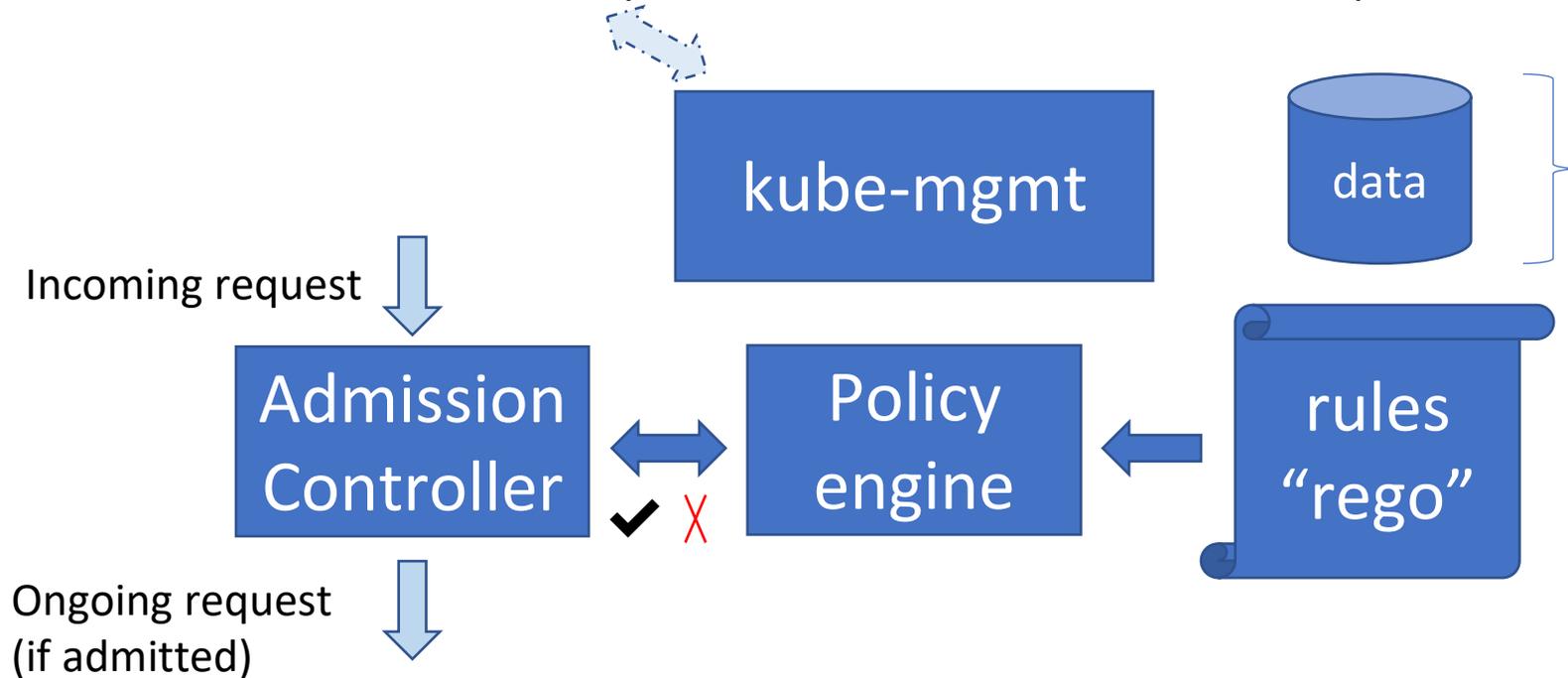




# Open Policy Agent (OPA)

## OPA & Kubernetes (GK v1)

Async data collection from Kubernetes api



Caching of resource data e.g. rules may act on pods but may require information about namespaces

Information about namespaces will not be in the incoming request but can be cached by OPA



# OPA – Policy Configuration

Create ConfigMaps containing Policies (Rego), e.g.

```
kubectl create configmap enforce-labels --from-file=enforce_labels.rego
```

Check status of ConfigMap

- annotation: status OK??



# Open Policy Agent Rules (Rego)

Example Rego policy rules

# Open Policy Agent Rules (Rego)

Enforce label Best Practices

# Open Policy Agent Rules (Rego)

## Enforce label Best Practices – Rego policy

```
operations={ "CREATE", "UPDATE" }

deny[msg] {
  input.request.kind.kind == "Deployment"

  # Check that this is one of the operations to apply Policy to:
  operations[ input.request.operation ]

  # Create the Sets of 'provided' and 'required' labels, then calculate
  provided := {label | input.request.object.metadata.labels[label]}
  required := {label | label := enforce_labels[_]}
  missing := required - provided
  count(missing) > 0
  msg := sprintf("Required labels are missing: %v", [missing])
}
```

```
enforce_labels = {
  "app.kubernetes.io/name",
  "app.kubernetes.io/instance",
  "app.kubernetes.io/version",
  "app.kubernetes.io/component",
  "app.kubernetes.io/part-of",
  "app.kubernetes.io/managed-by"
}
```

Adapted from example here:

<https://medium.com/@bouwe.ceunen/admission-control-on-kubernetes-9a1667b7e322>

# Open Policy Agent Rules (Rego)

## Enforce label Best Practices – input / output

INPUT (derived from OPA decision logs, here reduced):

```
{  "apiVersion": "admission.k8s.io/v1beta1",
  "kind": "AdmissionReview",
  "request": {
    "kind": { " kind ": " Deployment " },
    "operation": "CREATE",
    "object": {
      "apiVersion": "apps/v1",
      "kind": "Deployment",
      "metadata": {
        "labels": {
          "app": "ckad-demo-no-labels"
        },
        ....
      }
    }
  }
}
```

```
enforce_labels={
  "app.kubernetes.io/name",
}
```

OUTPUT (reduced):

```
{ "result": {
  "deny": [
    "Deployment/ckad-demo-no-labels: Required labels are missing:
    {\\"app.kubernetes.io/name\\"}"
  ],
  "enforce_labels": [
    "app.kubernetes.io/name",
    ...
  ],
}
```

Playground example: missing label - <https://play.openpolicyagent.org/p/lbbB2MfyXD>

# Open Policy Agent Rules (Rego)

## Enforce label Best Practices – input / output

INPUT (derived from OPA decision logs, here reduced):

```
{  "apiVersion": "admission.k8s.io/v1beta1",
  "kind": "AdmissionReview",
  "request": {
    "kind": { " kind ": " Deployment " },
    "operation": "CREATE",
    "object": {
      "apiVersion": "apps/v1",
      "kind": "Deployment",
      "metadata": {
        "labels": {
          "app": "ckad-demo-with-labels",
          "app.kubernetes.io/name": "ckad-demo-with-labels"
        }
      }
    }
  }
}
```

```
enforce_labels={
  "app.kubernetes.io/name",
}
```

OUTPUT (full):

```
{ "result": {
  "deny": [],
  "enforce_labels": [
    "app.kubernetes.io/name"
  ],
  "operations": [
    "CREATE", "UPDATE"
  ]
}
```

....

Playground example: missing label - <https://play.openpolicyagent.org/p/lbbB2MfyXD>

Policy as Code / Policy Engines  
Open Policy Agent & eco-system  
Specialized tools: Kyverno  
Summary





# Kyverno



# Kyverno - Configuration

Policies in Kyverno are defined completely as YAML resources

They can be

- Namespaced of kind *Policy*
- or cluster-wide of kind *ClusterPolicy*

A policies may be:

- Validating or auditing
- Mutating
- Generating



# Kyverno - Installation

## Helm:

```
helm repo add kyverno https://kyverno.github.io/kyverno/  
helm repo update  
helm install kyverno kyverno/kyverno --namespace kyverno --create-namespace
```

Or directly from YAML:

```
kubectl create -f  
https://raw.githubusercontent.com/kyverno/kyverno/main/definitions/release/install.  
yaml
```



# Kyverno - Examples

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: scenario1-require-labels
spec:
  validationFailureAction: enforce
  rules:
  - name: check-for-labels
    match:
      resources:
        kinds:
        - Pod
    validate:
      message: "[enforce] kyverno says -label `app.kubernetes.io/name` is required"
      pattern:
        metadata:
          labels:
            app.kubernetes.io/name: "?*"
```



# Kyverno - Examples

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: scenario1-require-labels
spec:
  validationFailureAction: enforce
  rules:
  - name: check-for-labels
    match:
      resources:
        kinds:
          - Pod
    validate:
      message: "[enforce] kyverno says -label `app.kubernetes.io/name` is required"
      pattern:
        metadata:
          labels:
            app.kubernetes.io/name: "?*"
```

Policy as Code / Policy Engines  
Open Policy Agent & eco-system  
Specialized tools: Kyverno

Summary



# Summary – Comparison OPA-GK/Kyverno

Both target Kubernetes using native resources and can replace PSP

OPA/GK: since 2017

- Rego allows complex rules – but steeper learning curve (Rego)
- “OPA” Allows to manage full IT stack, beyond Kubernetes
- HA

Kyverno: since 2019

- More familiar YAML for rules
- Generation capabilities

Choice is great ...

# Summary – Comparison OPA-GK/Kyverno

Features/Capabilities	Gatekeeper	Kyverno
Validation	✓	✓
Mutation	✓* (alpha)	✓
Generation	✗	✓
Policy as native resources	✓	✓
Metrics exposed	✓	✗
OpenAPI validation schema (kubectl explain)	✗	✓
High Availability	✓	✗
API object lookup	✓	✓* (alpha)
CLI with test ability	✓** (separate CLI)	✓
Policy audit ability	✓	✓

# Summary

- Policy engines provide a way to dynamically provide policy rules
- Policy engines may be
  - open source or not
  - declarative or not
  - platform-specific or not
- Kubernetes can be extended using policy engines  
(needed due to impending PSP deprecation)

OPA/GK and Kyverno are useful options, other engines exist.

# Thank you !



 <https://linkedin.com/in/mjbright>

 @mjbright

 @mjbright

@mjbright CONSULTING

# Resources

Resource	URL
	Kubernetes Admission Control
Kubernetes Blog	A Guide to Kubernetes Admission Controllers   Kubernetes <a href="https://kubernetes.io/blog/2019/03/21/a-guide-to-kubernetes-admission-controllers/">https://kubernetes.io/blog/2019/03/21/a-guide-to-kubernetes-admission-controllers/</a>
Banzai Cloud Blog	In-depth introduction to Kubernetes admission webhooks · Banzai Cloud <a href="https://banzaicloud.com/blog/k8s-admission-webhooks/">https://banzaicloud.com/blog/k8s-admission-webhooks/</a>
Kubernetes Documentation	Dynamic Admission Control <a href="https://kubernetes.io/docs/reference/access-authn-authz/extensible-admission-controllers/">https://kubernetes.io/docs/reference/access-authn-authz/extensible-admission-controllers/</a>

# Resources

Resource	URL
	Open Policy Agent
OPA website	<a href="https://www.openpolicyagent.org/">https://www.openpolicyagent.org/</a>
OPA docs	<a href="https://www.openpolicyagent.org/docs/latest/">https://www.openpolicyagent.org/docs/latest/</a>
Styra Academy	<a href="https://academy.styra.com/">https://academy.styra.com/</a>
OPA blog	<a href="https://blog.openpolicyagent.org/">https://blog.openpolicyagent.org/</a>
Rego Playground	<a href="https://play.openpolicyagent.org/">https://play.openpolicyagent.org/</a>

# Resources

Resource	URL
	Open Policy Agent
OPA github	<a href="https://github.com/open-policy-agent/opa">https://github.com/open-policy-agent/opa</a>
OPA gatekeeper github	<a href="https://github.com/open-policy-agent/gatekeeper">https://github.com/open-policy-agent/gatekeeper</a>
Conftest github	<a href="https://github.com/open-policy-agent/conftest">https://github.com/open-policy-agent/conftest</a>
VScode Extension	<a href="https://github.com/open-policy-agent/vscode-opa">https://github.com/open-policy-agent/vscode-opa</a>

# Resources

Resource	URL
	Kyverno
Kyverno website	<a href="https://kyverno.io/">https://kyverno.io/</a>
Kyverno github	<a href="https://github.com/kyverno/kyverno/">https://github.com/kyverno/kyverno/</a>

# Resources

Resource	URL
	OPA & Kyverno
2021-Feb Chip Zoller	Kubernetes Policy Comparison: OPA/Gatekeeper vs Kyverno <a href="https://neonmirrors.net/post/2021-02/kubernetes-policy-comparison-opa-gatekeeper-vs-kyverno/">https://neonmirrors.net/post/2021-02/kubernetes-policy-comparison-opa-gatekeeper-vs-kyverno/</a>

# Thank you !



<https://linkedin.com/in/mjbright>



@mjbright



@mjbright

@mjbright CONSULTING