Policy-based access control

An introduction to Open Policy Agent

Anders Eknert

- Developer advocate at Styra
- Software development
- Background in identity systems
- Two years into OPA
- Cooking and food
- Football

@anderseknert

anderseknert

Challenge:

Manage policy in increasingly distributed, complex and heterogeneous systems

Challenge:

Manage policy in increasingly distributed, complex and heterogeneous systems

Challenge:

Manage policy in increasingly distributed, complex and heterogeneous systems

python Java

Groovy

THE C PROGRAMMING LANGUAGE

php

kubernetes docker HashiCorp Terraform

Google Cloud

Azure

aws

IBM Cloud

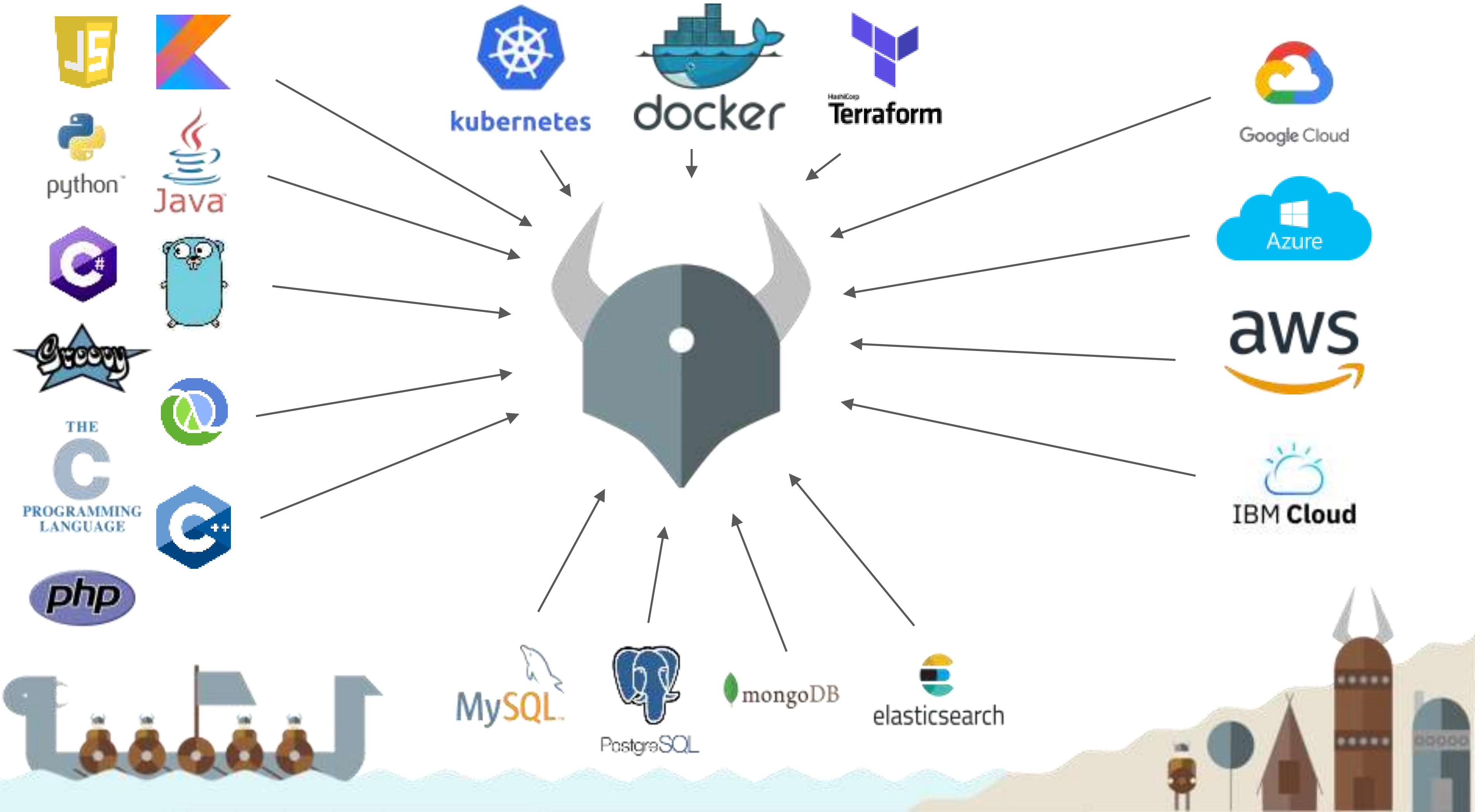
Challenge:

Manage policy in increasingly distributed, complex and heterogeneous systems

Challenge:

Manage policy in increasingly distributed, complex and heterogeneous systems

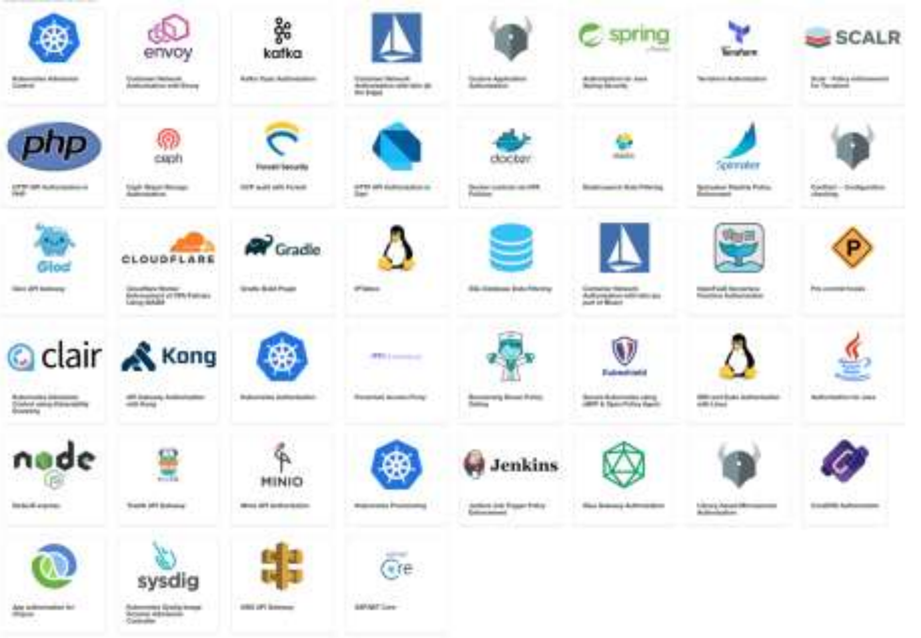
Goal:
Unify policy enforcement across the stack

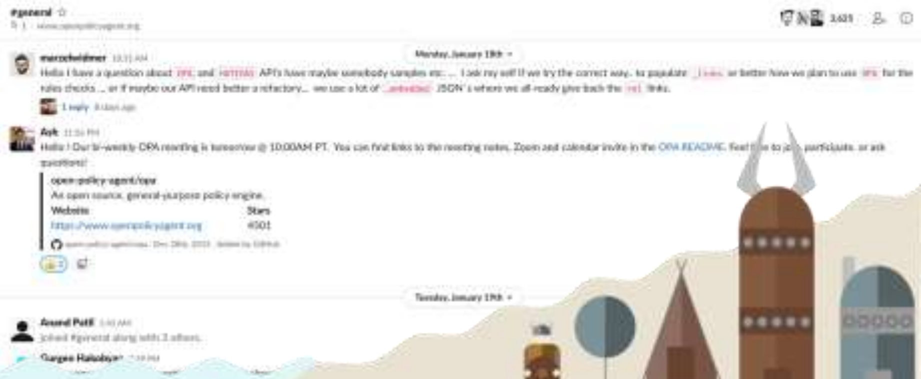

JS
python
Java
Groovy
THE C PROGRAMMING LANGUAGE
php
kubernetes
docker
HashiCorp
Terraform
Google Cloud
Azure
aws
IBM Cloud
MySQL
PostgreSQL
mongoDB
elasticsearch

- Open source general purpose policy engine
- Unified toolset and framework for policy across the stack
- Decouples policy from application logic
- Separates policy *decision* from *enforcement*
- Policies written in declarative language Rego
- Popular use cases ranging from kubernetes admission control, microservice authorization, infrastructure, data source filtering, to CI/CD pipeline policies and many more.

Vibrant community

- 160 contributors
- 50+ integrations
- 4500+ Github Stars
- 3600+ Slack users
- 30+ million Docker image pulls
- Ecosystem including Conftest, Gatekeeper, VS Code and IntelliJ editor plugins.




#general ☆

www.openpolicyagent.org

3,625

Monday, January 18th

marcoschwidmer 10:31 AM
Hello I have a question about [illegible] and [illegible] API's have maybe somebody samples etc ... I ask my self if we try the correct way.. to populate [illegible] or better how we plan to use [illegible] for the rules checks ... or if maybe our API need better a refactory... we use a lot of _embedded JSON's where we all-ready give back the rel links.

1 reply [illegible]

Ash 11:56 PM
Hello ! Our bi-weekly OPA meeting is tomorrow @ 10:00AM PT. You can find links to the meeting notes, Zoom and calendar invite in the OPA README. Feel free to join, participate, or ask questions!

open-policy-agent/opa
An open source, general-purpose policy engine.

Website	Stars
https://www.openpolicyagent.org	4501

open-policy-agent/opa | Dec 29th, 2020 | Added by GitHub

Tuesday, January 19th

Anand Patil [illegible] AM
joined #general along with 3 others.

Gurgen Hakobyan [illegible] PM

Production users

NETFLIX

Azure

Goldman Sachs

Pinterest

reddit

ATLASSIAN

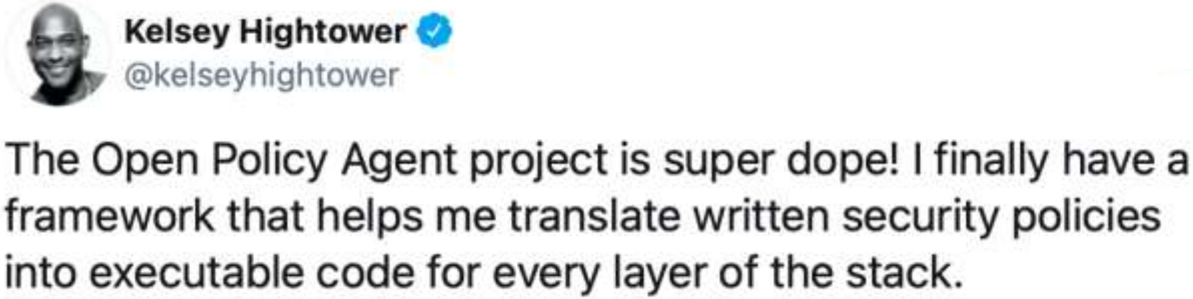
CapitalOne

CLOUDFLARE

yelp

CHEF

intuit

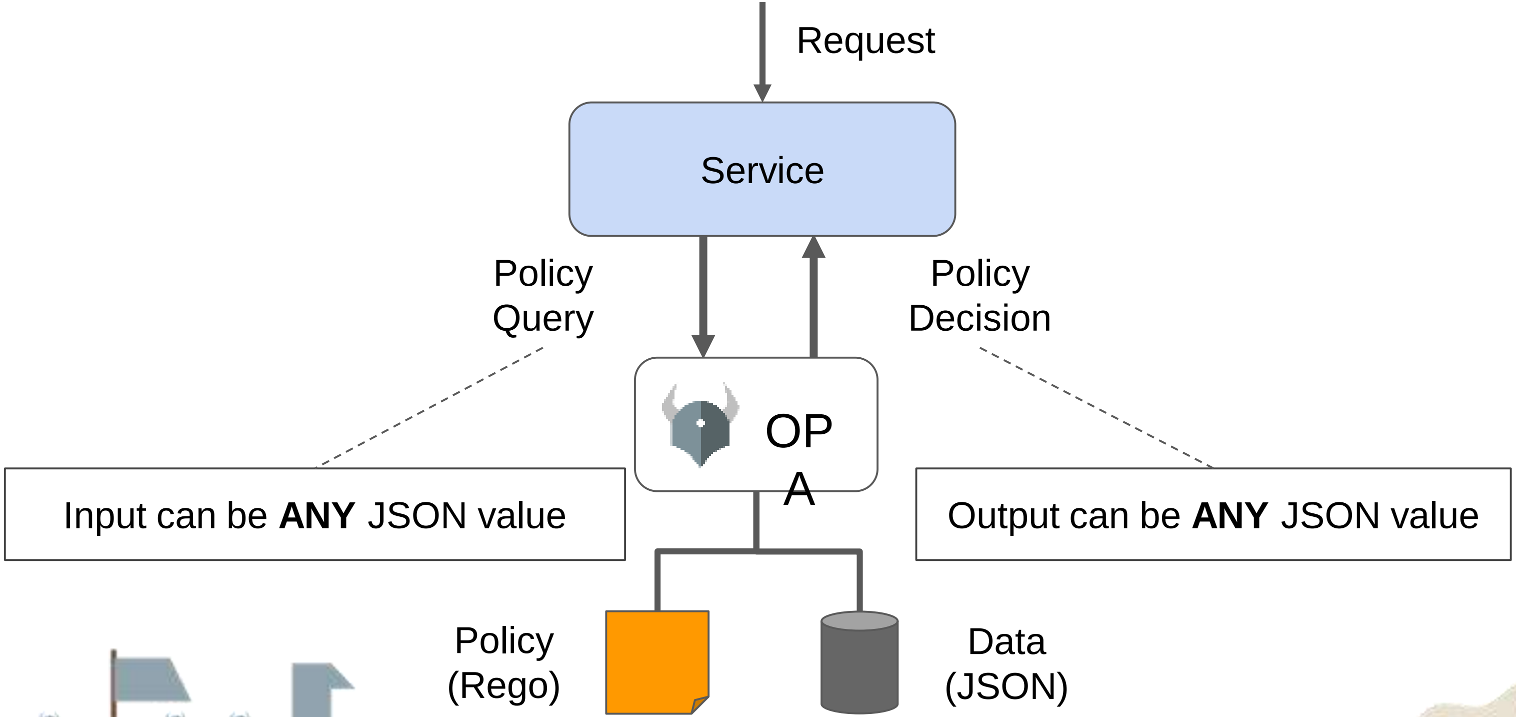


Kelsey Hightower
@kelseyhightower

The Open Policy Agent project is super dope! I finally have a framework that helps me translate written security policies into executable code for every layer of the stack.

OPA and Rego

Policy decision model



Request

Service

Policy
Query

Policy
Decision

OP
A

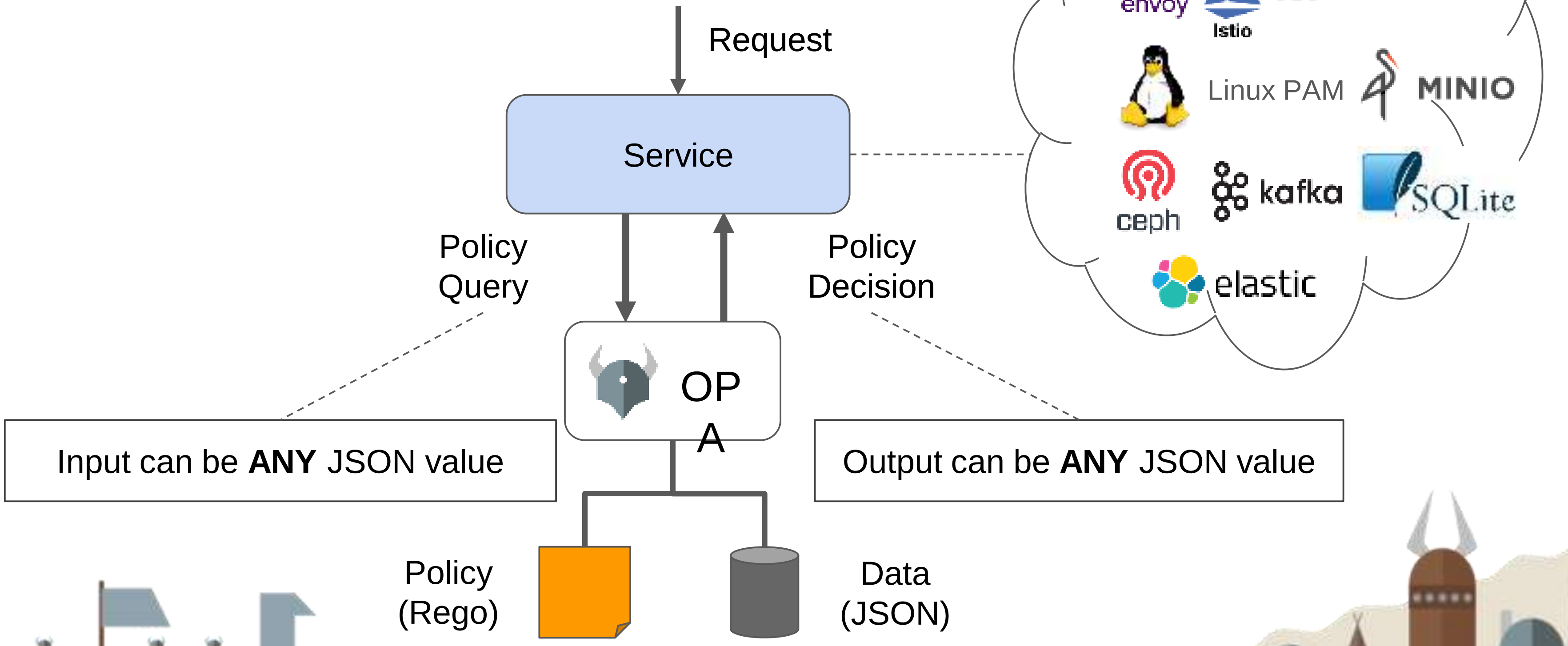
Input can be **ANY** JSON value

Output can be **ANY** JSON value

Policy
(Rego)

Data
(JSON)

Policy decision model

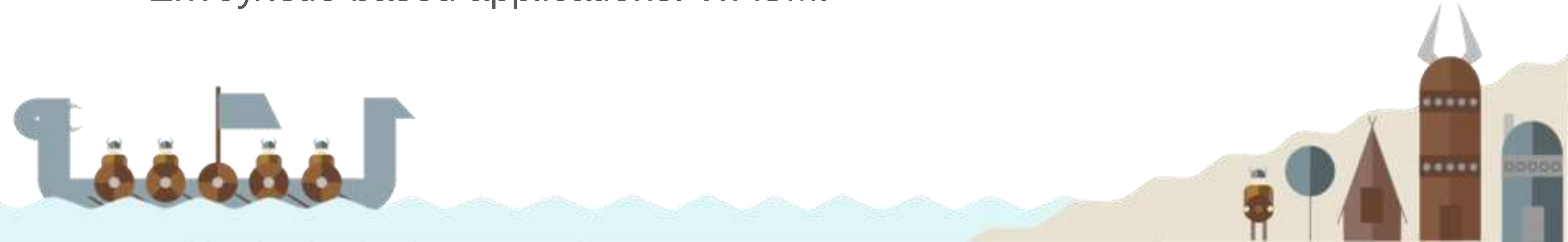


Input can be **ANY** JSON value

Output can be **ANY** JSON value

Deployment

- OPA runs as a lightweight self-contained server binary.
- OPA ideally deployed as close to service as possible. This usually means on the same host, as a daemon or in a sidecar container.
- Applications communicate with the OPA server through its REST API.
- Go library available for Go applications.
- Envoy/Istio based applications. WASM.



Policy authoring and Rego

- Declarative high-level policy language used by OPA.
- Policy consists of any number of rules.
- Rules commonly return true/false but may return any type available in JSON, like strings, lists and objects.
- 140+ built-in functions: JWTs, date/time, CIDR math ,etc.
- Policy testing is easy with provided unit test framework.
- Well documented! https://www.openpolicyagent.org/docs/latest/
- Try it out! https://play.openpolicyagent.org/

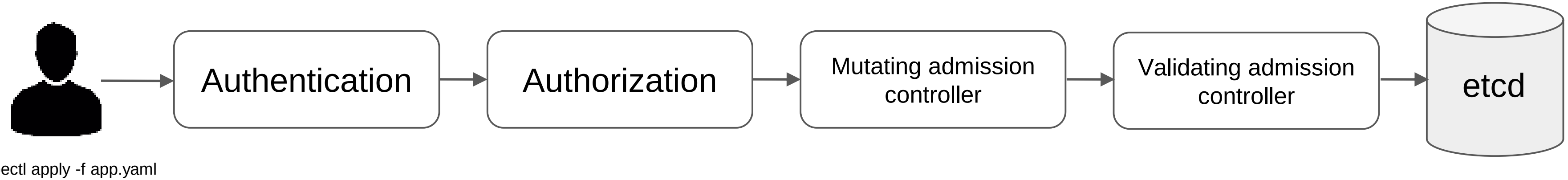
Policy data

- JSON Web Tokens
- As part of query input
- Push data
- Bundle API
- http.send function from inside policy

Demo

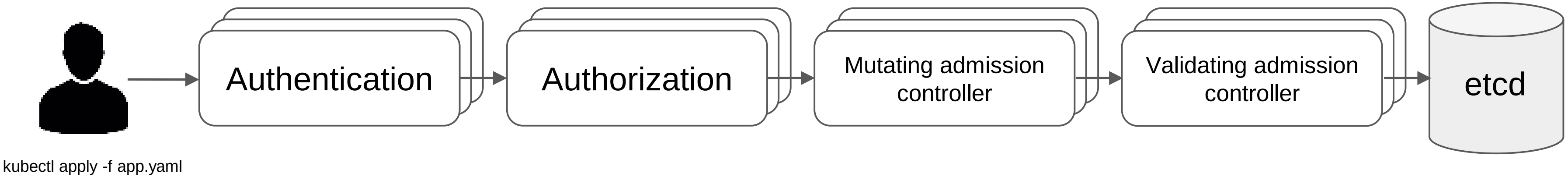
Kubernetes

The Kubernetes API


Authentication
Authorization
Mutating admission controller
Validating admission controller
etcd
kubectl apply -f app.yaml

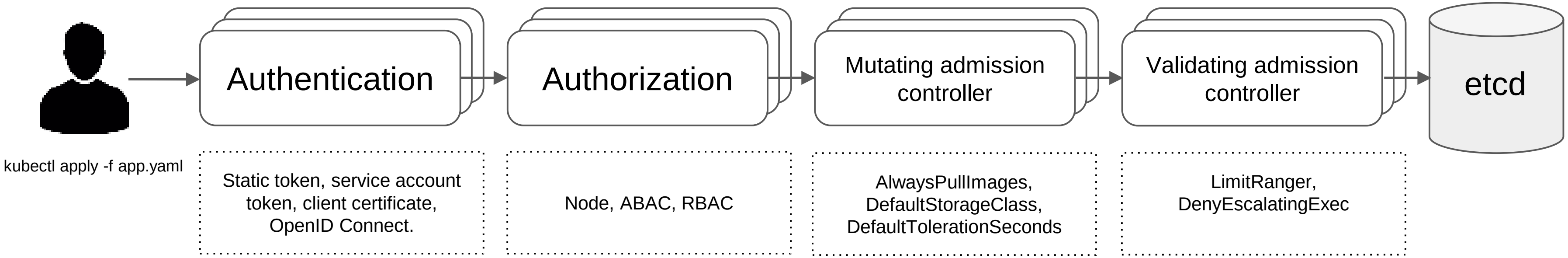
Before a resource is persisted in etcd must first pass a series of modules

The Kubernetes API


Authentication
Authorization
Mutating admission controller
Validating admission controller
etcd
kubectl apply -f app.yaml

Modules are chainable

The Kubernetes API



kubectl apply -f app.yaml

Authentication

Authorization

Mutating admission controller

Validating admission controller

etcd

Static token, service account token, client certificate, OpenID Connect.

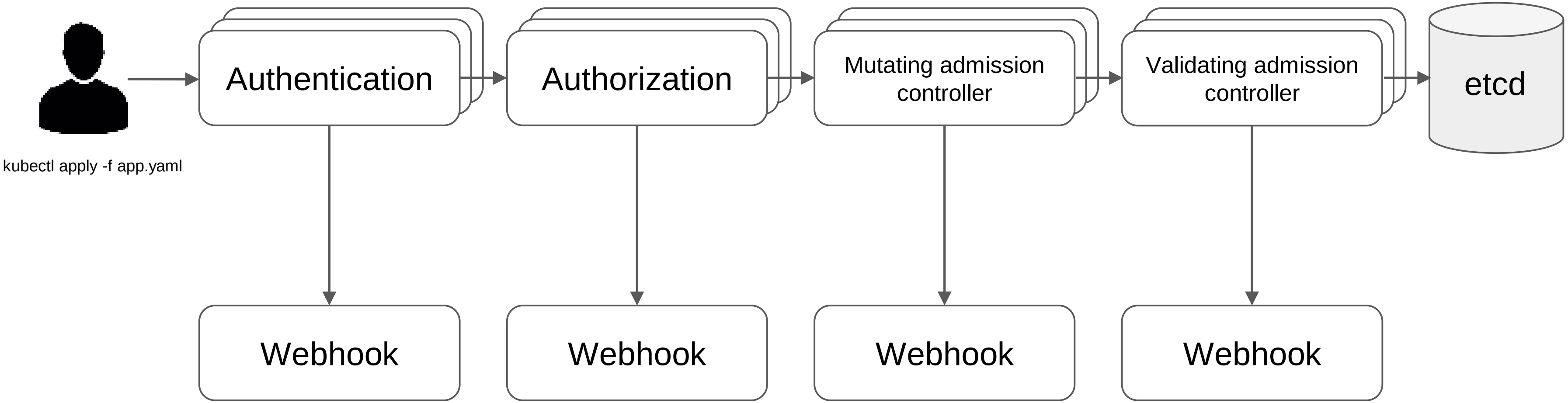
Node, ABAC, RBAC

AlwaysPullImages, DefaultStorageClass, DefaultTolerationSeconds

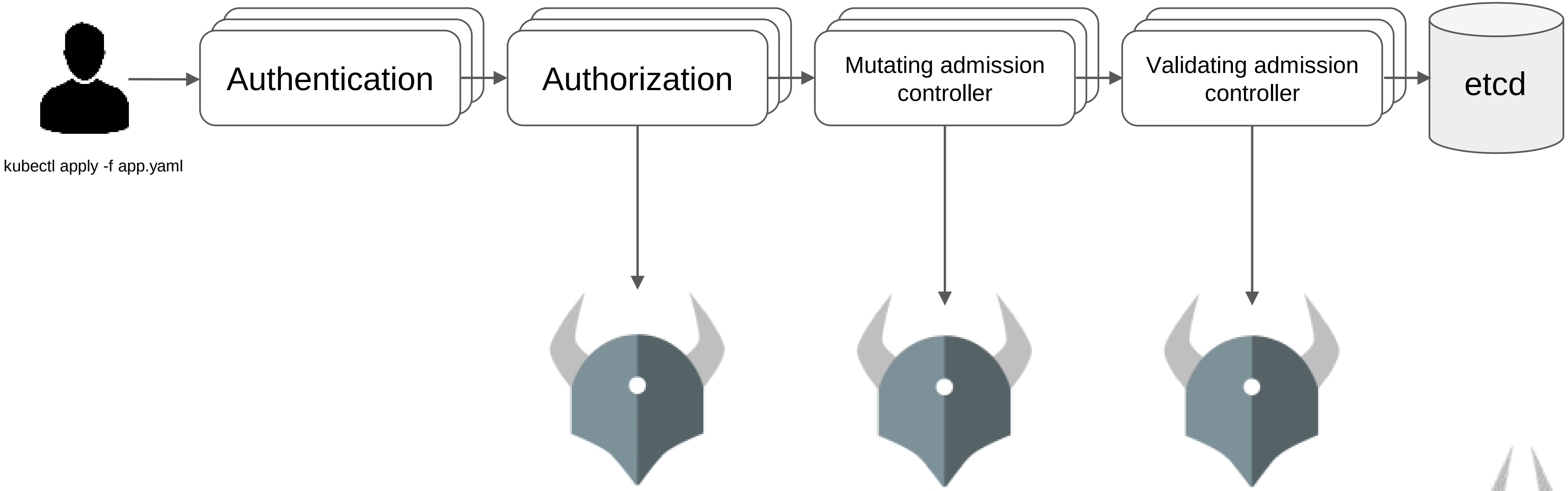
LimitRanger, DenyEscalatingExec

Built-in modules

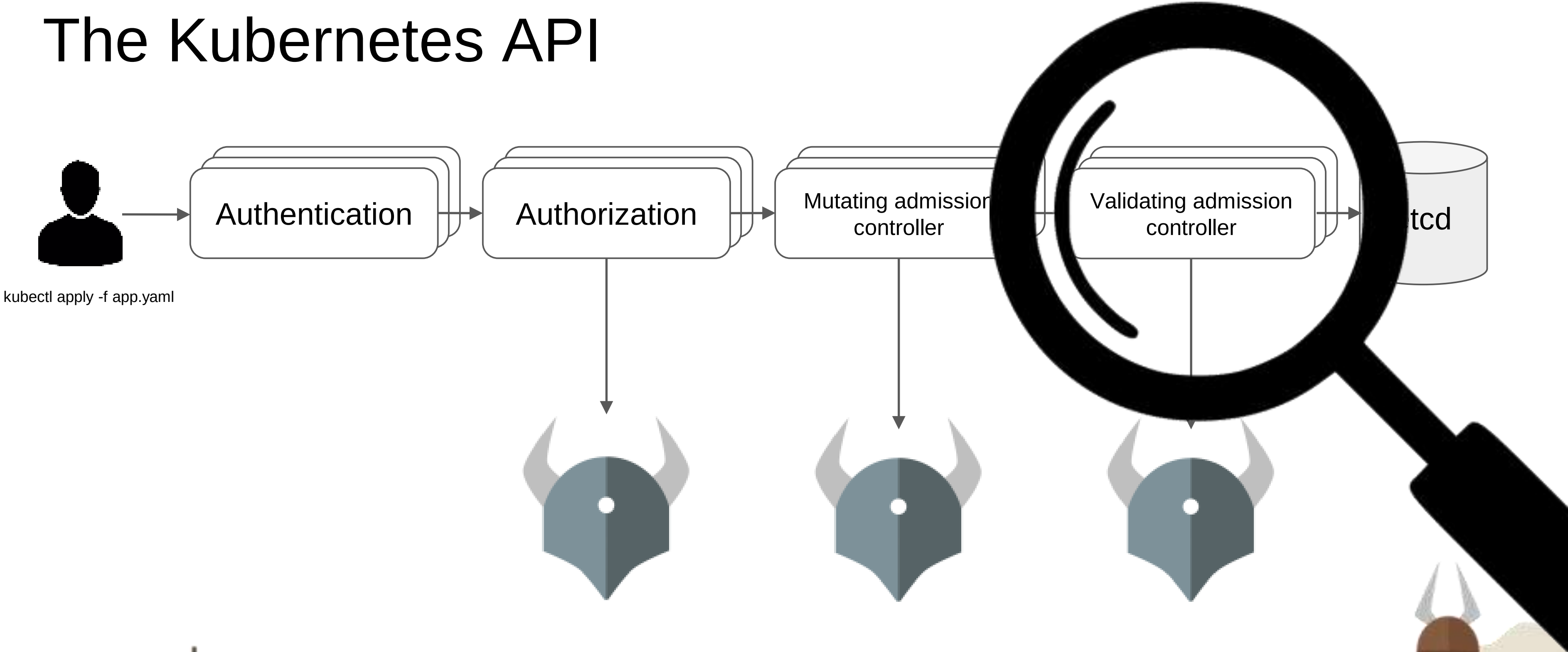
The Kubernetes API


kubectl apply -f app.yaml
Authentication
Authorization
Mutating admission controller
Validating admission controller
etcd
Webhook
Webhook
Webhook
Webhook

The Kubernetes API


Authentication
Authorization
Mutating admission controller
Validating admission controller
etcd
kubectl apply -f app.yaml

The Kubernetes API



kubectl apply -f app.yaml

Authentication

Authorization

Mutating admission controller

Validating admission controller

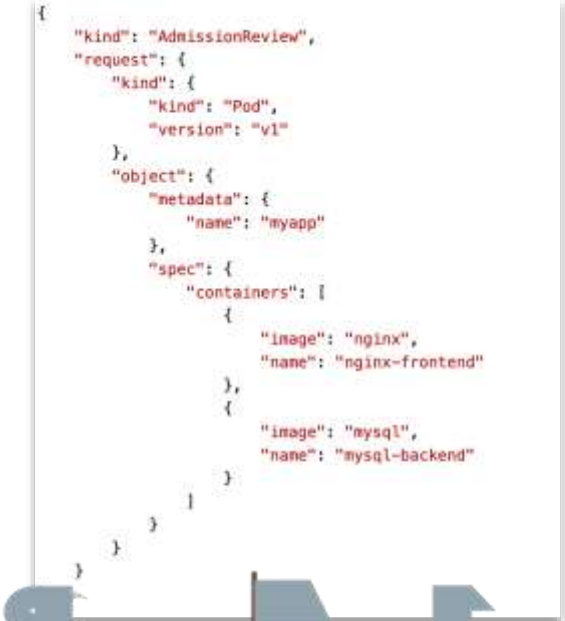
etcd

Validating admission controller

- By far the most popular module to extend
- Allows building policy-based guardrails around clusters
- Common policies enforce:
  - Use of internal Docker registry and other image constraints
  - Required labels on resources - team belonging, cost centre, etc
  - Ingress host/path uniqueness
  - HTTPS for services
  - Deny attributes like hostPath volume mounts
  - Limits on resource allocation
  - Pod Security Policies
  - ...anything really

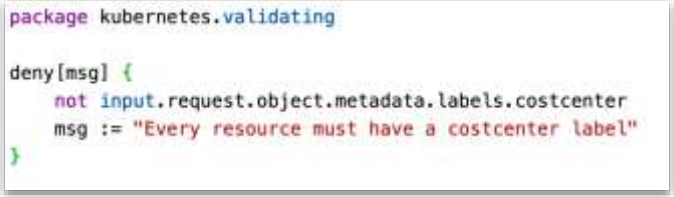
Kubernetes validating admission controller webhook

Input



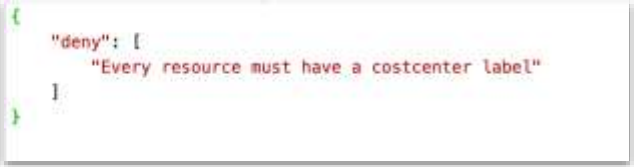
```
{
    "kind": "AdmissionReview",
    "request": {
        "kind": {
            "kind": "Pod",
            "version": "v1"
        },
        "object": {
            "metadata": {
                "name": "myapp"
            },
            "spec": {
                "containers": [
                    {
                        "image": "nginx",
                        "name": "nginx-frontend"
                    },
                    {
                        "image": "mysql",
                        "name": "mysql-backend"
                    }
                ]
            }
        }
    }
}
```

Policy



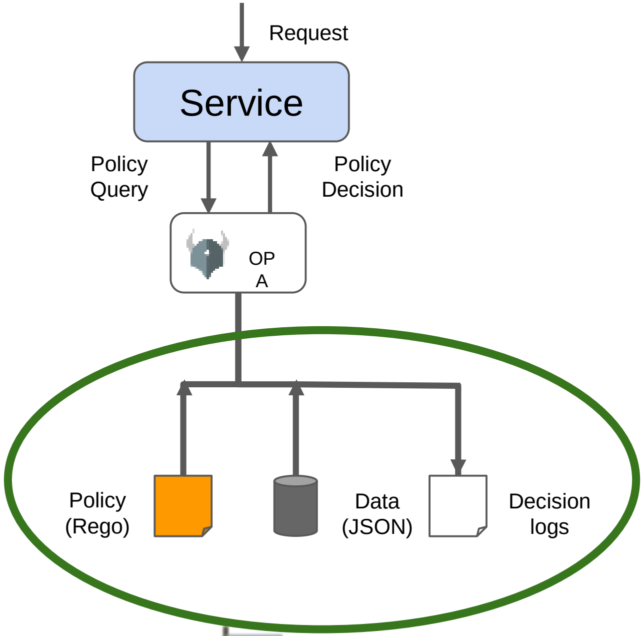
```
package kubernetes.validating

deny[msg] {
    not input.request.object.metadata.labels.costcenter
    msg := "Every resource must have a costcenter label"
}
```

Output



```
{
    "deny": [
        "Every resource must have a costcenter label"
    ]
}
```

Management APIs



Managing OPA at scale

- Bundle API - distribute policy and data from a central location
- Decision log API - allow OPA instances to report back on any decisions made. This may be used for auditing as well as for refinement of policies.
- Status API - allows OPA to send status and health updates to the management server.
- Discovery API - provides OPA instances the option to periodically fetch configuration.

Getting started

- Start small – write a few simple policies and tests.
- Browse the OPA documentation. Get a feel for the basics and the built-ins.
- Consider possible applications near to you - previous apps and libraries you’ve worked with. Consider the informal policies it dealt with.
- Delegate policy responsibilities to OPA. Again, start small! Perhaps a single endpoint to begin somewhere. Deploy and build experience.
- Scale up - consider management, logging, bundle server, etc.
- Styra Academy - https://academy.styra.com
- Styra DAS - http://www.styra.com/das-free
- Join the OPA Slack community! - https://openpolicyagent.slack.com

Thank you!