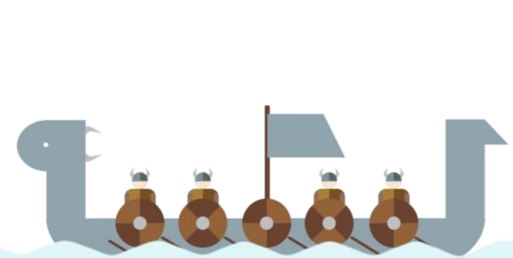
Open Policy Agent (OPA)

Introduction - KubeCon Shanghai 2019



Torin Sandall

Engineer at Styra
Co-creator of OPA

`tsandall on OPA slack`

`sometorin`

`openpolicyagent.org`

OPA: Community

Inception

Project started in 2016 at Styra.

Goal

Unify policy enforcement across the stack.

Users

Netflix
Chef
Medallia
Cloudflare
State Street
Pinterest
Intuit
Capital One
...and many more.

Use Cases

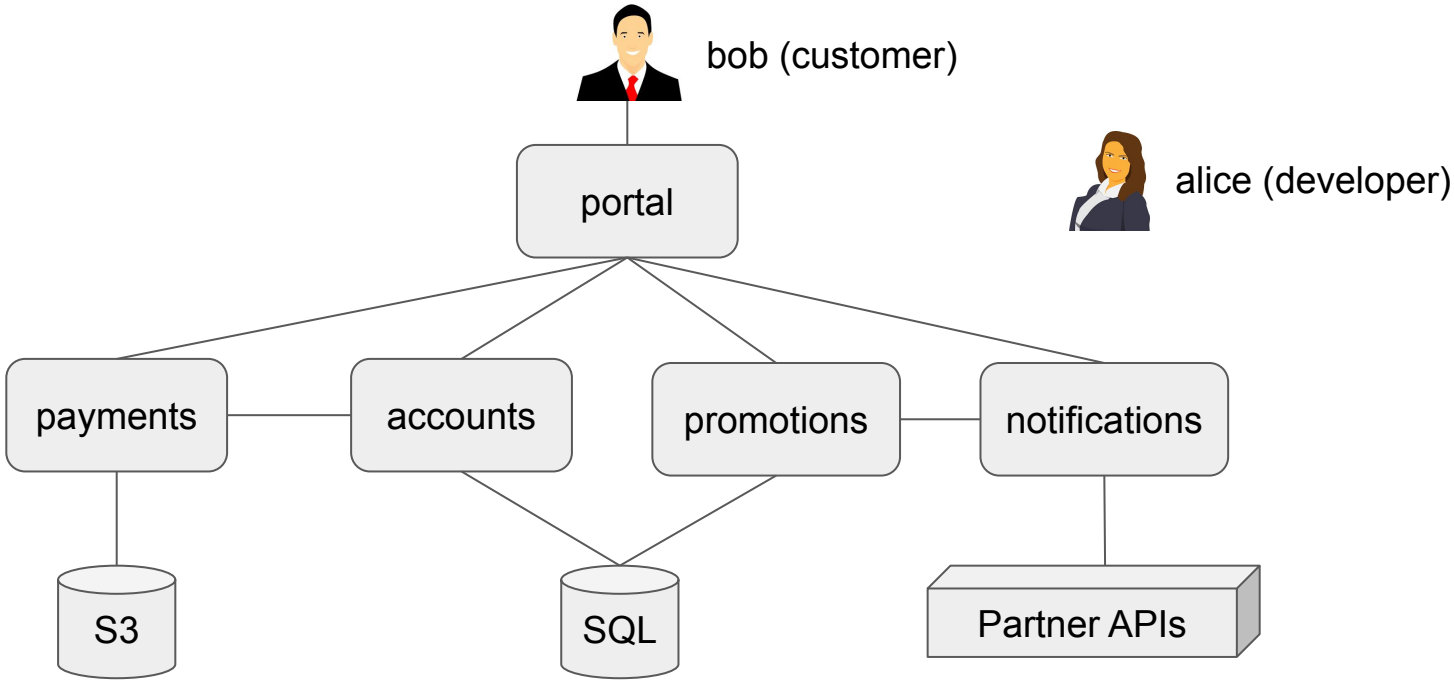
Admission control
Authorization
  ACLs
  RBAC
  IAM
  ABAC
Risk management
Data Protection
Data Filtering

Today

CNCF project (Incubating)

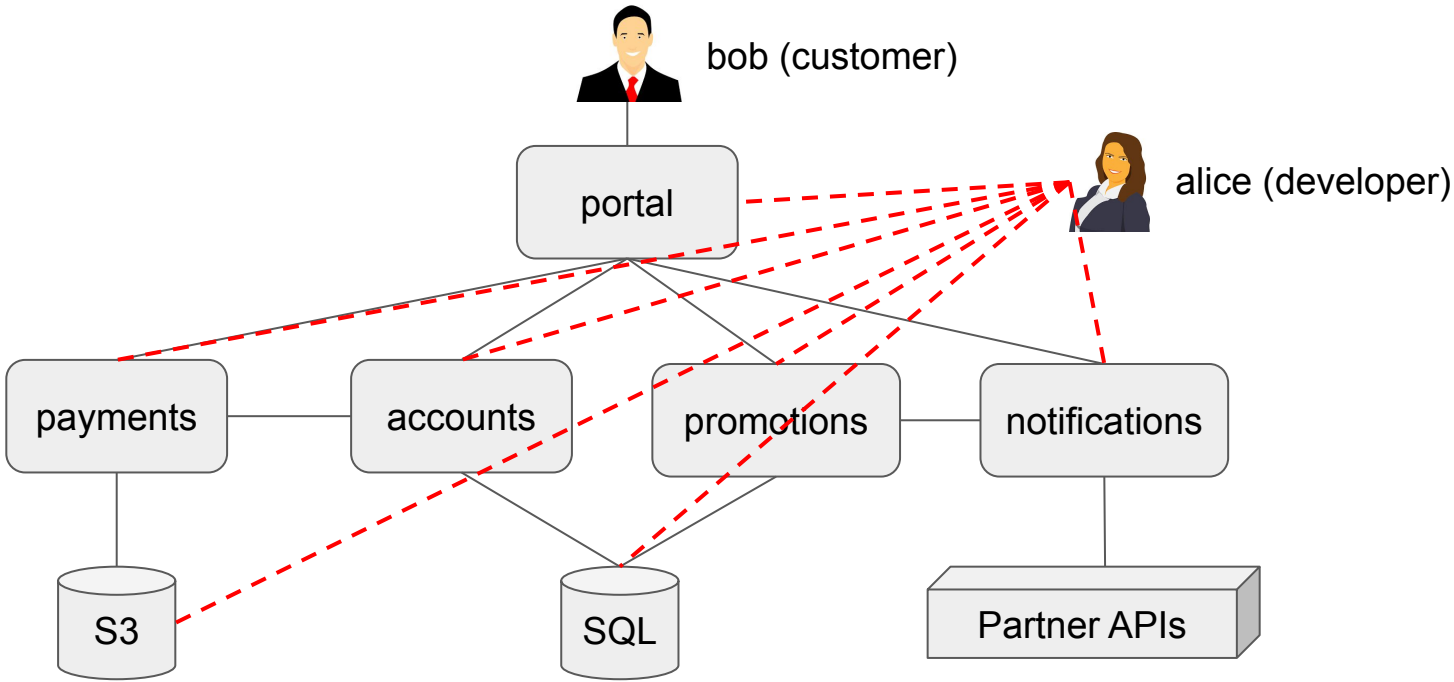
60+ contributors
900+ slack members
2,000+ stars
20+ integrations

Example: Application



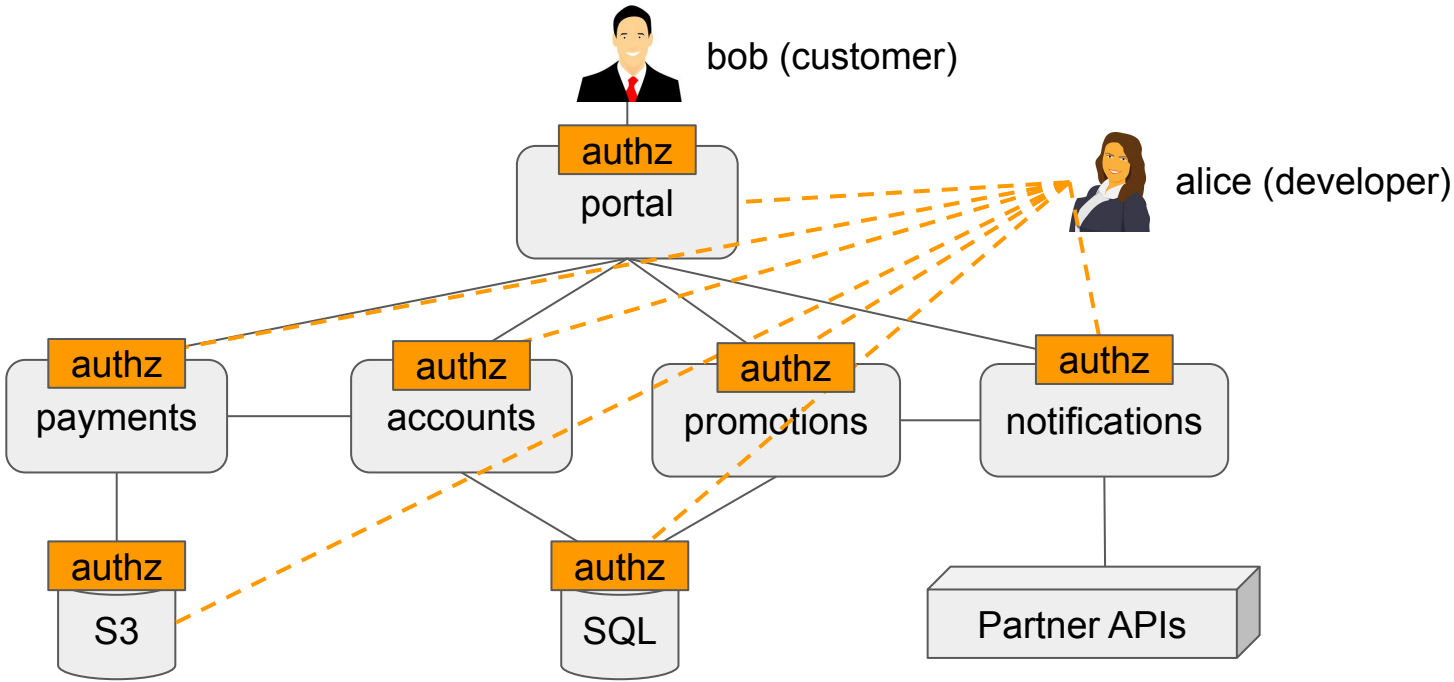
openpolicyagent.org

Example: Application



openpolicyagent.org

Example: Application

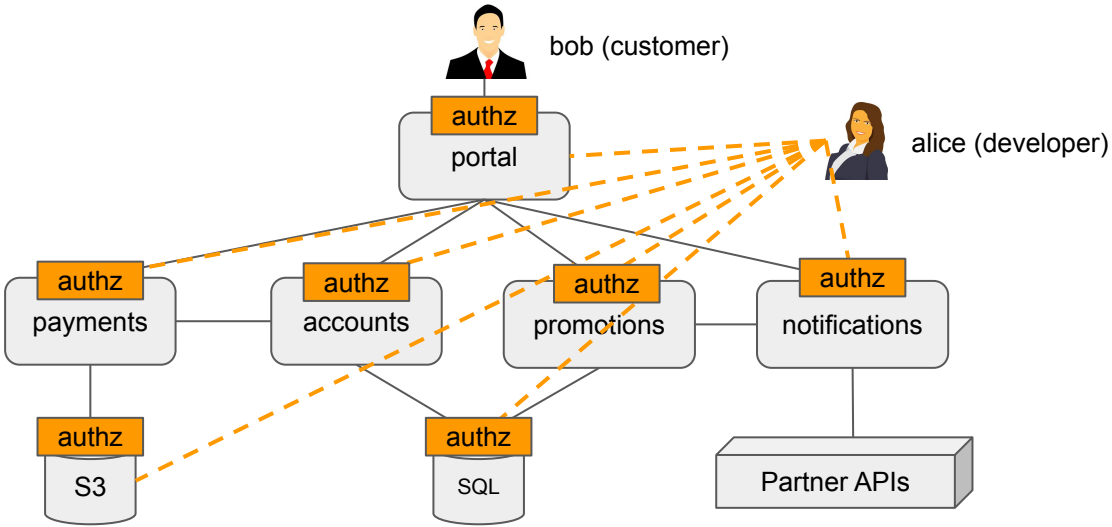


openpolicyagent.org

Example: Application

Obvious questions...

- How do you enforce new policies from infosec, compliance, or legal?
- How do you delegate control to your end-users?
- How do you roll-out policy changes?
- How do you leverage context, e.g., HR/User DB?
- How do you render UIs based on policy?
- How do you test your policies for correctness?
- What about 100+ services written in Java, Ruby, ...



openpolicyagent.org

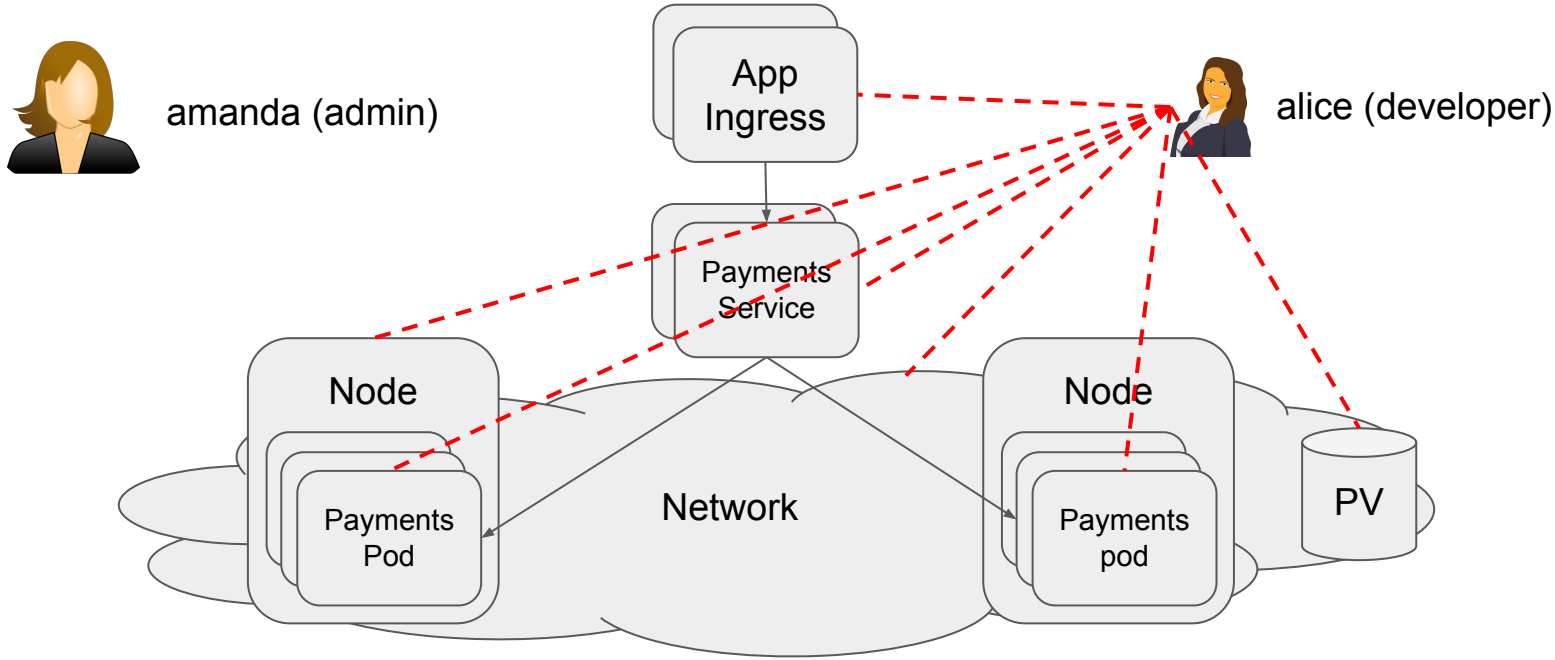
Example: Kubernetes Platform

amanda (admin)

alice (developer)

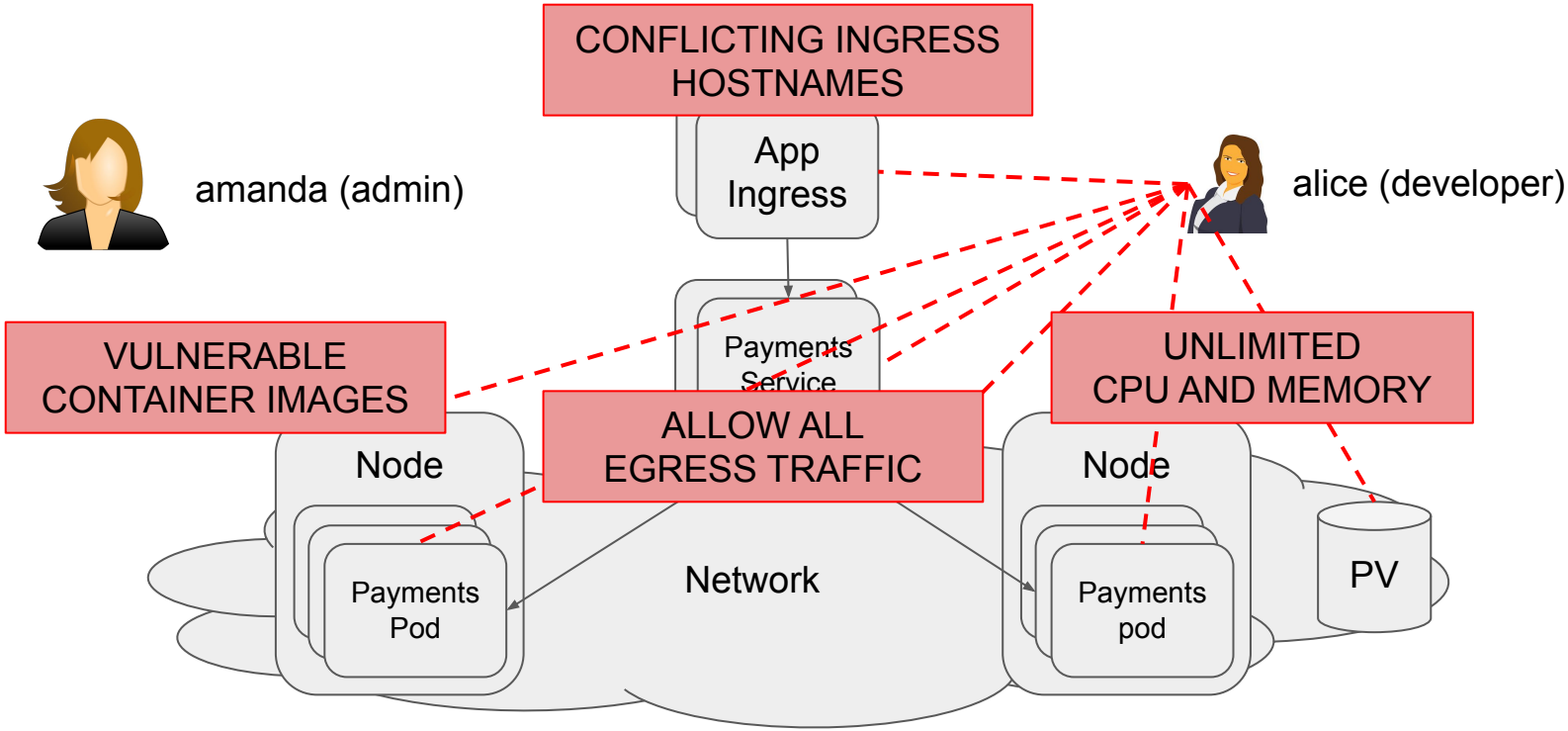
openpolicyagent.org

Example: Kubernetes Platform



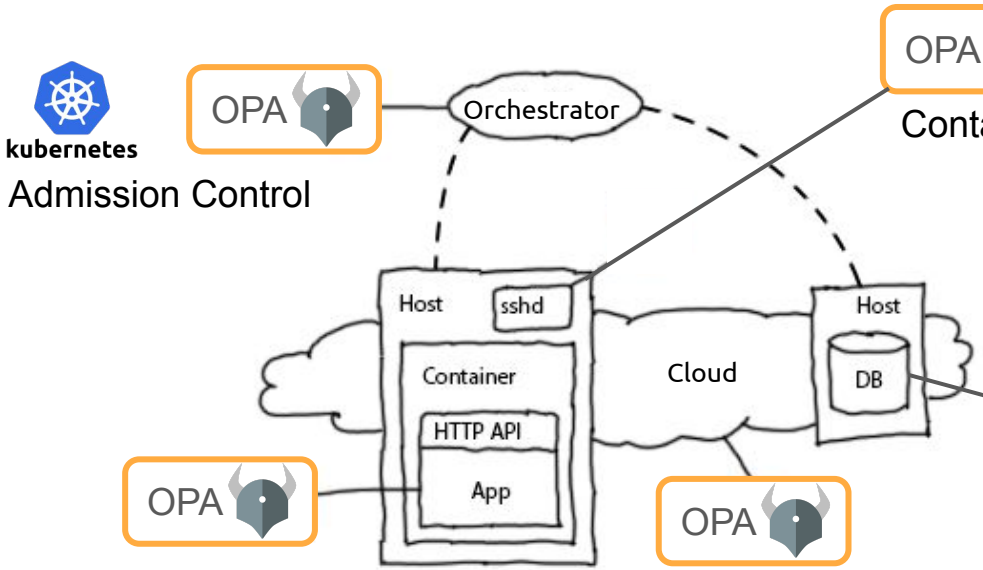
openpolicyagent.org

Example: Kubernetes Platform



openpolicyagent.org

OPA: Unified Policy Enforcement Across the Stack



kubernetes

Admission Control

OPA

Container Execution, SSH, sudo

docker

Linux

OPA

Data Protection & Data Filtering

ceph

kafka

MINIO

SQLite

elastic

OPA

Microservice APIs

OPA

Risk Management

Istio

LINKERD

spring

Kong

envoy

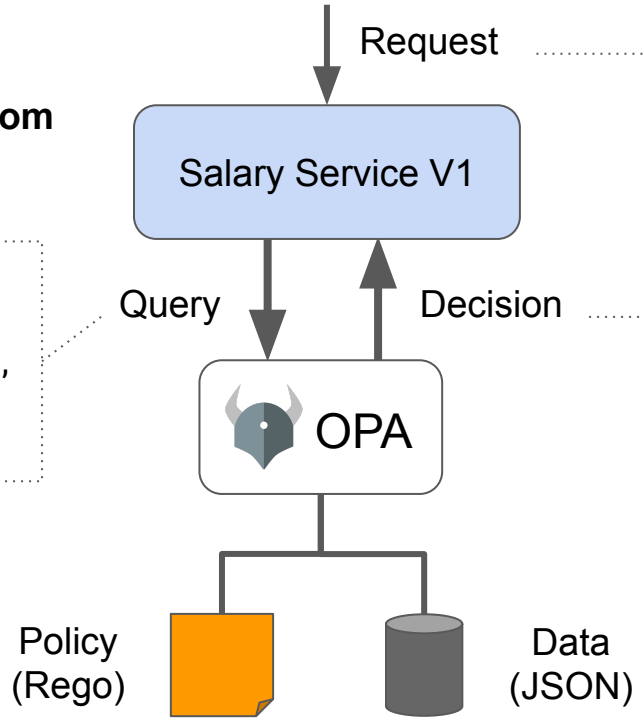
Terraform

Forseti Security

openpolicyagent.org

OPA: General-purpose Policy Engine

Enforcement is decoupled from decision-making.




```
GET /salary/bob HTTP/1.1
Authorization: alice
```

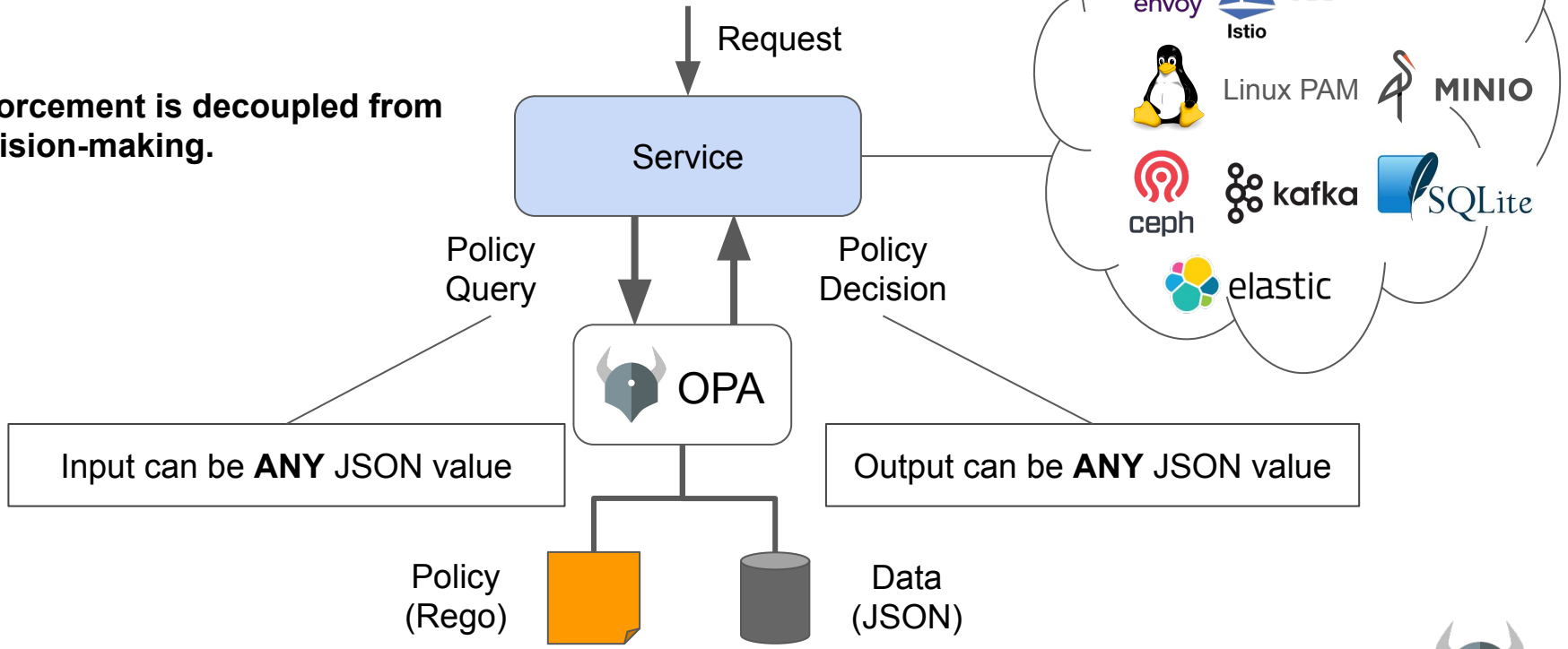
```
{
  "method": "GET",
  "path": ["salary", "bob"],
  "user": "alice"
}
```

```
true or false
```

openpolicyagent.org

OPA: General-purpose Policy Engine

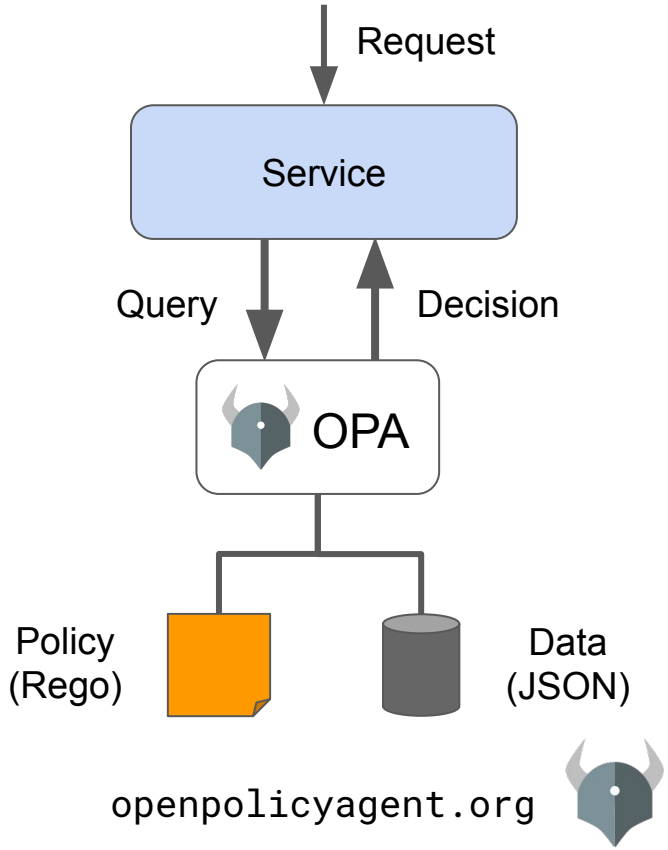
Enforcement is decoupled from decision-making.



openpolicyagent.org

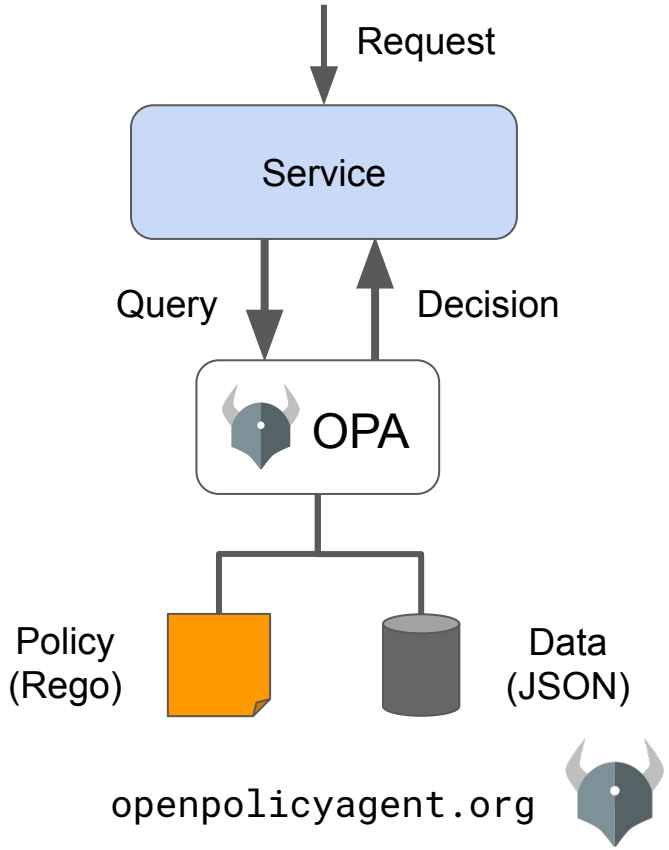
OPA: Features

- **Declarative Policy Language (Rego)**
  - Can user X do operation Y on resource Z?
  - What invariants does workload W violate?
  - Which records should bob be allowed to see?



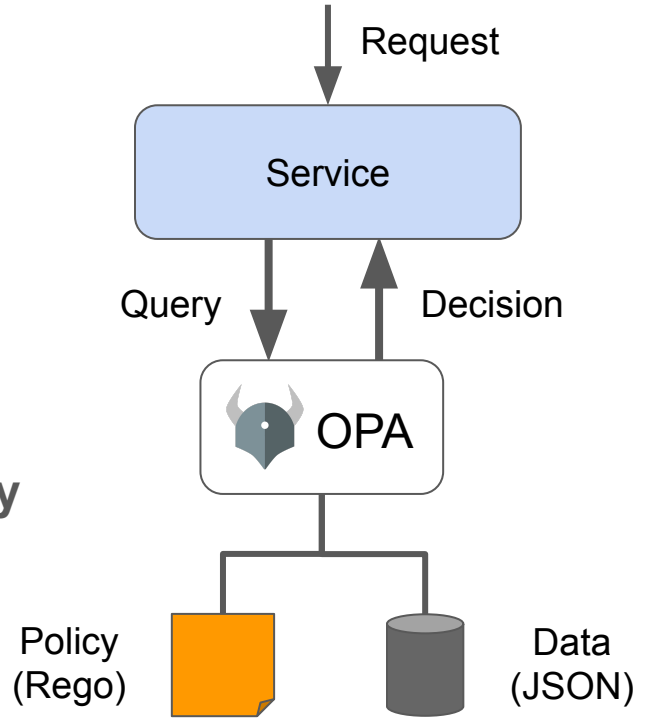
OPA: Features

- **Declarative Policy Language (Rego)**
  - Can user X do operation Y on resource Z?
  - What invariants does workload W violate?
  - Which records should bob be allowed to see?
- **Library (Go), sidecar/host-level daemon**
  - Policy and data are kept in-memory
  - Zero decision-time dependencies



OPA: Features

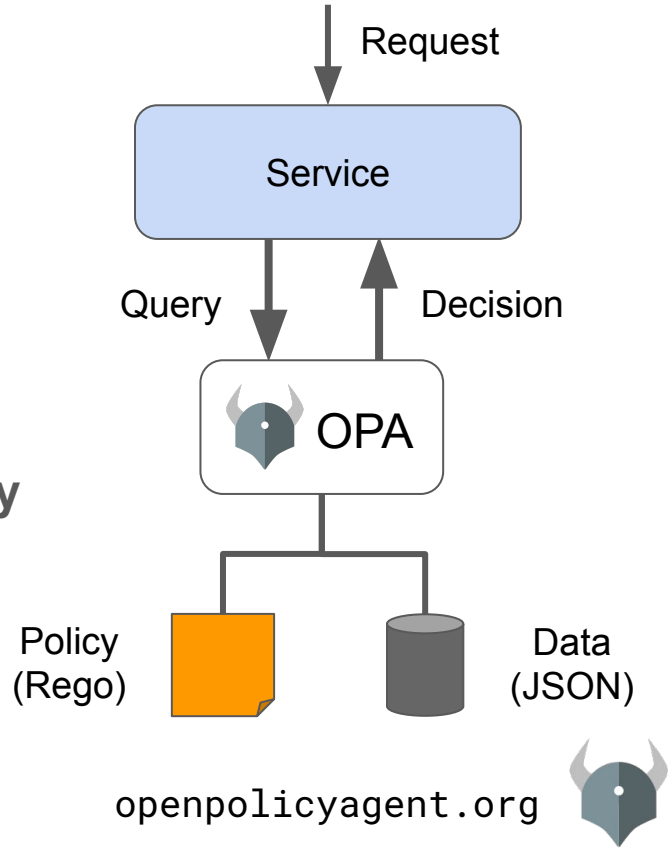
- **Declarative Policy Language (Rego)**
  - Can user X do operation Y on resource Z?
  - What invariants does workload W violate?
  - Which records should bob be allowed to see?
- **Library (Go), sidecar/host-level daemon**
  - Policy and data are kept in-memory
  - Zero decision-time dependencies
- **Management APIs for control & observability**
  - Bundle service API for sending policy & data to OPA
  - Status service API for receiving status from OPA
  - Log service API for receiving audit log from OPA



openpolicyagent.org

OPA: Features

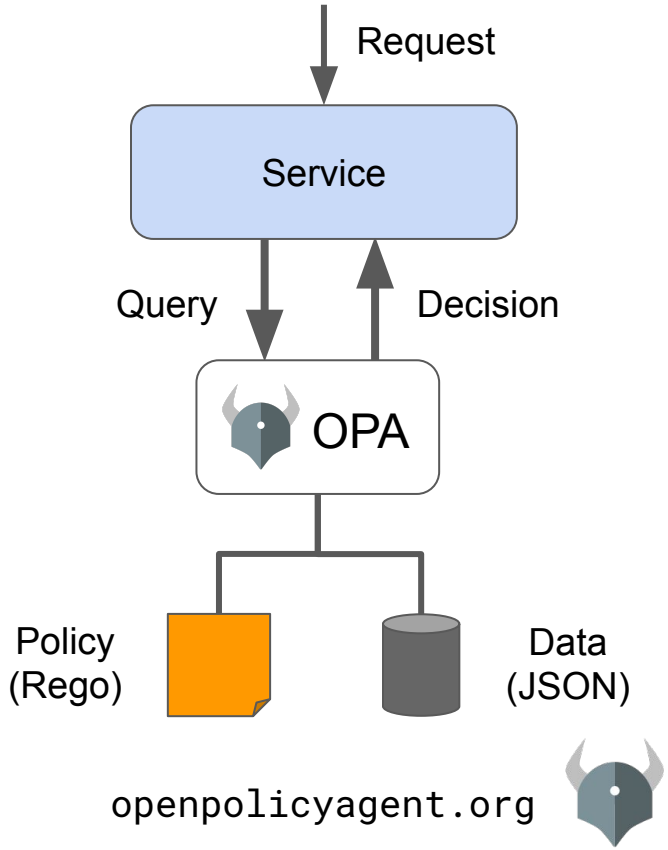
- **Declarative Policy Language (Rego)**
  - Can user X do operation Y on resource Z?
  - What invariants does workload W violate?
  - Which records should bob be allowed to see?
- **Library (Go), sidecar/host-level daemon**
  - Policy and data are kept in-memory
  - Zero decision-time dependencies
- **Management APIs for control & observability**
  - Bundle service API for sending policy & data to OPA
  - Status service API for receiving status from OPA
  - Log service API for receiving audit log from OPA
- **Tooling to build, test, and debug policy**
  - opa run, opa test, opa fmt, opa deps, opa check, etc.
  - VS Code plugin, Tracing, Profiling, etc.



OPA: Example

Example policy

"Employees can read their own salary and the salary of anyone they manage."



OPA: Integrations

	Admission Control	*“Restrict ingress hostnames for payments team.”* *“Ensure container images come from corporate repo.”*
	API Authorization	*“Deny test scripts access to production services.”* *“Allow analysts to access APIs serving anonymized data.”*
 Linux PAM	SSH & sudo	*“Only allow on-call engineers to SSH into production servers.”*
	Data Protection	*"Trades exceeding $10M must be executed between 9AM and 5PM and require MFA."*
	Data Filtering	*"Users can access files for past 6 months related to the region they licensed."*

`openpolicyagent.org`

NETFLIX Use Case: Authorization

Uses OPA to enforce **access control** in microservices across a **variety of languages and frameworks** for **thousands of instances** in their cloud infrastructure. Netflix takes advantage of OPA's ability to bring in **contextual information** and data from remote resources in order to evaluate policies in a **flexible and consistent manner**. For a description of how Netflix has architected access control with OPA check out this talk from KubeCon Austin 2017.

Use Case: API Authorization

Integrates OPA to implement **IAM-style access control** and **enumerate user->resource permissions** in Chef Automate V2. The integration utilizes OPA's Partial Evaluation feature to reduce evaluation time (in exchange for higher update latency.)

Use Case: k8s Admission Control

Uses OPA as a **validating and mutating admission controller** to implement various security, multi-tenancy, and risk management policies across approximately **50 clusters and 1,000 namespaces**. For more information on how Intuit uses OPA see this talk from KubeCon Seattle 2018.

Thank You! Questions?

Tuesday, June 25 • 18:15 - 18:50

Gatekeeper: Flexible, Shareable Policy for Kubernetes - Craig Peters, Mircosoft

Deep Dive: Kubernetes Policy WG - Zhipeng Huang, Huawei

open-policy-agent/opa

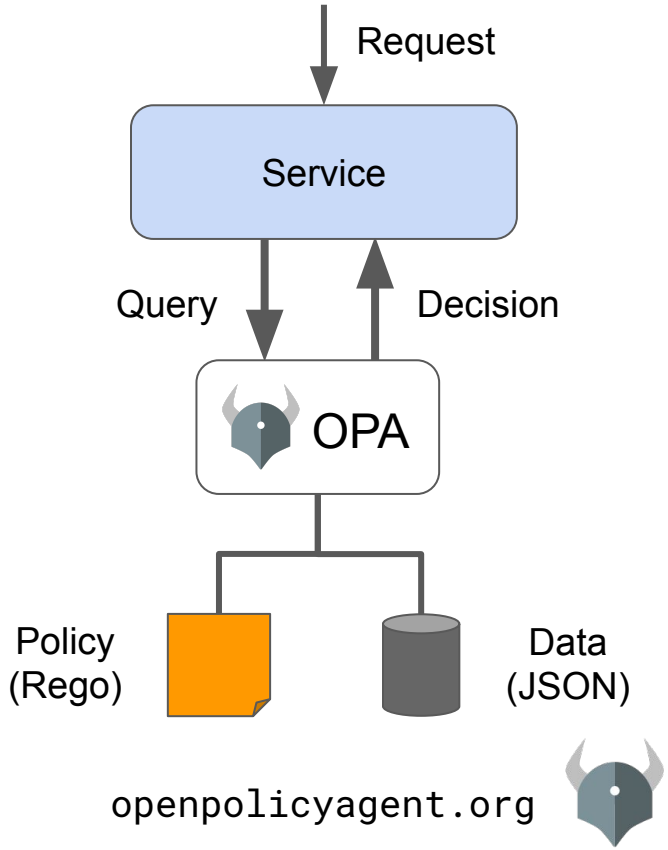
slack.openpolicyagent.org

openpolicyagent.org

How Does OPA Work?

Example policy

"Employees can read their own salary and the salary of anyone they manage."



OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "bob"
```

openpolicyagent.org

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "bob"
```

```
allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = employee_id
}
```

openpolicyagent.org

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "bob"
```

```
allow = true {
  input.method = "GET"
  input.path = ["salary", "bob"]
  input.user = "bob"
}
```

openpolicyagent.org

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

```
allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = employee_id
}
```

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "alice"
```

openpolicyagent.org

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "alice"
```

Different user now!

```
allow = true {
  input.method = "GET"
  input.path = ["salary", "bob"]
  input.user = "bob"
}
```

This statement will "FAIL"

openpolicyagent.org

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "alice"
```

Context Data

```
{
  "managers": {
    "bob": ["alice", "fred"]
    "alice": ["fred"]
  }
}
```

```
allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = employee_id
}
```

openpolicyagent.org

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "alice"
```

Context Data

```
{
  "managers": {
    "bob": ["alice", "fred"]
    "alice": ["fred"]
  }
}
```

```
import data.managers

allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = employee_id
}

allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = managers[employee_id][_]
}
```

openpolicyagent.org

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "alice"
```

Context Data

```
{
  "managers": {
    "bob": ["alice", "fred"]
    "alice": ["fred"]
  }
}
```

```
import data.managers

allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = employee_id
}

allow = true {
  input.method = "GET"
  input.path = ["salary", "bob"]
  input.user = managers["bob"][_]
}
```

openpolicyagent.org

OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

```
method: "GET"
path: ["salary", "bob"]
user: "alice"
```

Context Data

```
{
  "managers": {
    "bob": ["alice", "fred"]
    "alice": ["fred"]
  }
}
```

```
import data.managers

allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = employee_id
}

allow = true {
  input.method = "GET"
  input.path = ["salary", "bob"]
  input.user = "alice"
}
```

openpolicyagent.org

OPA: Declarative Language (Rego)

More information at openpolicyagent.org

See **How Do I Write Policies?**
- Explains language constructs

See **Language Reference**
- Documents built-in functions: glob, regex, JWTs, x509, etc.

See **Tutorials** section
- HTTP APIs, Kubernetes, Docker, Terraform, Kafka, SSH, etc.

```
import data.managers

allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = employee_id
}

allow = true {
  input.method = "GET"
  input.path = ["salary", "bob"]
  input.user = "alice"
}
```

openpolicyagent.org