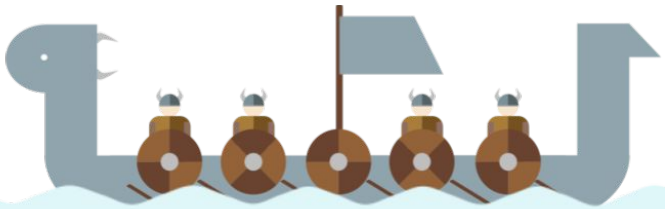
Open Policy Agent (OPA)

Unified Cloud-native Policy Control



Who Are We?

Tim Hinrichs

Co-founder & CTO at Styra
Co-creator of OPA

@tim on OPA
@tlhinrichs

Torin Sandall

Engineer at Styra
Co-creator of OPA

@tsandall on OPA
@sometorin

openpolicyagent.org

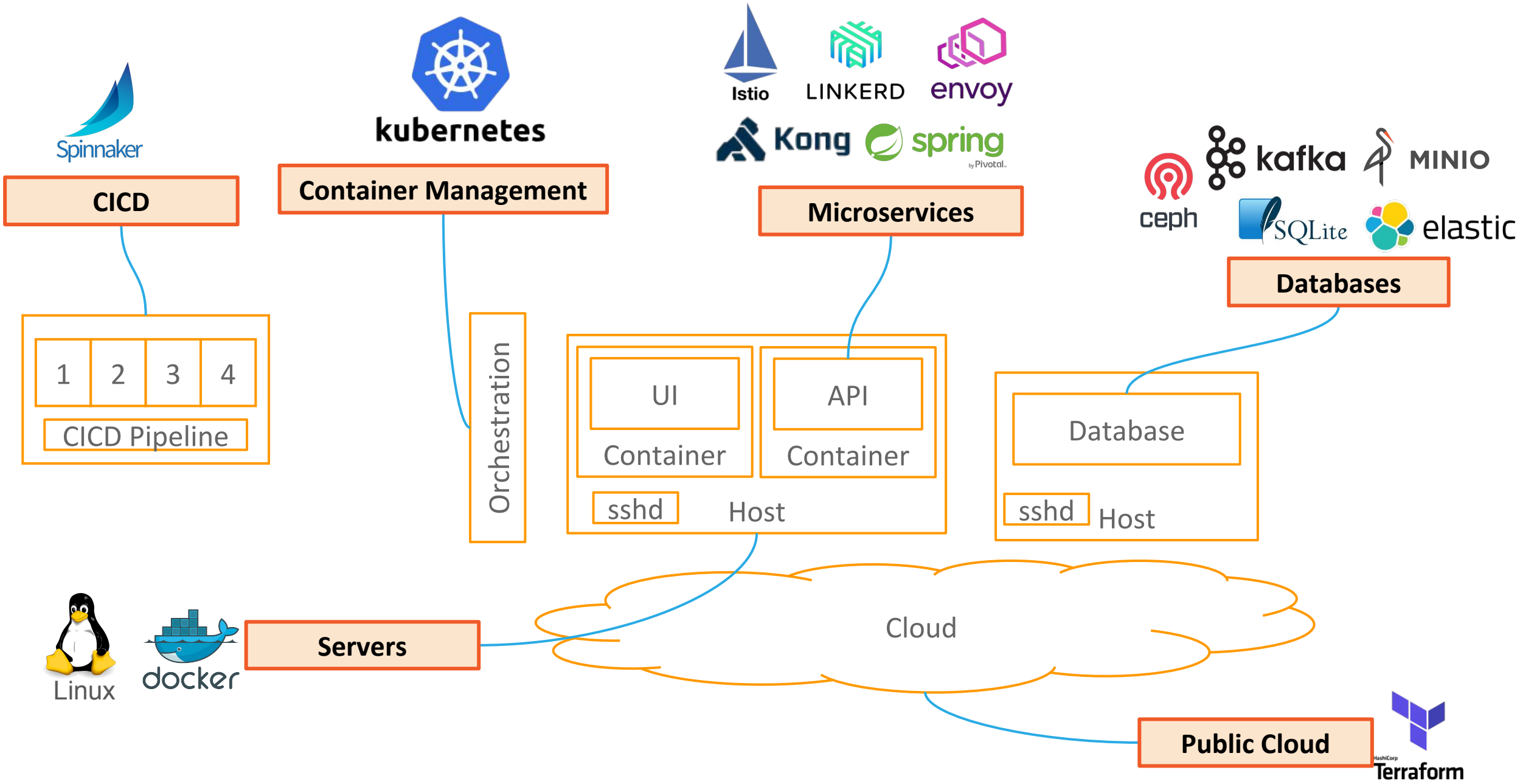
Agenda

OPA Overview

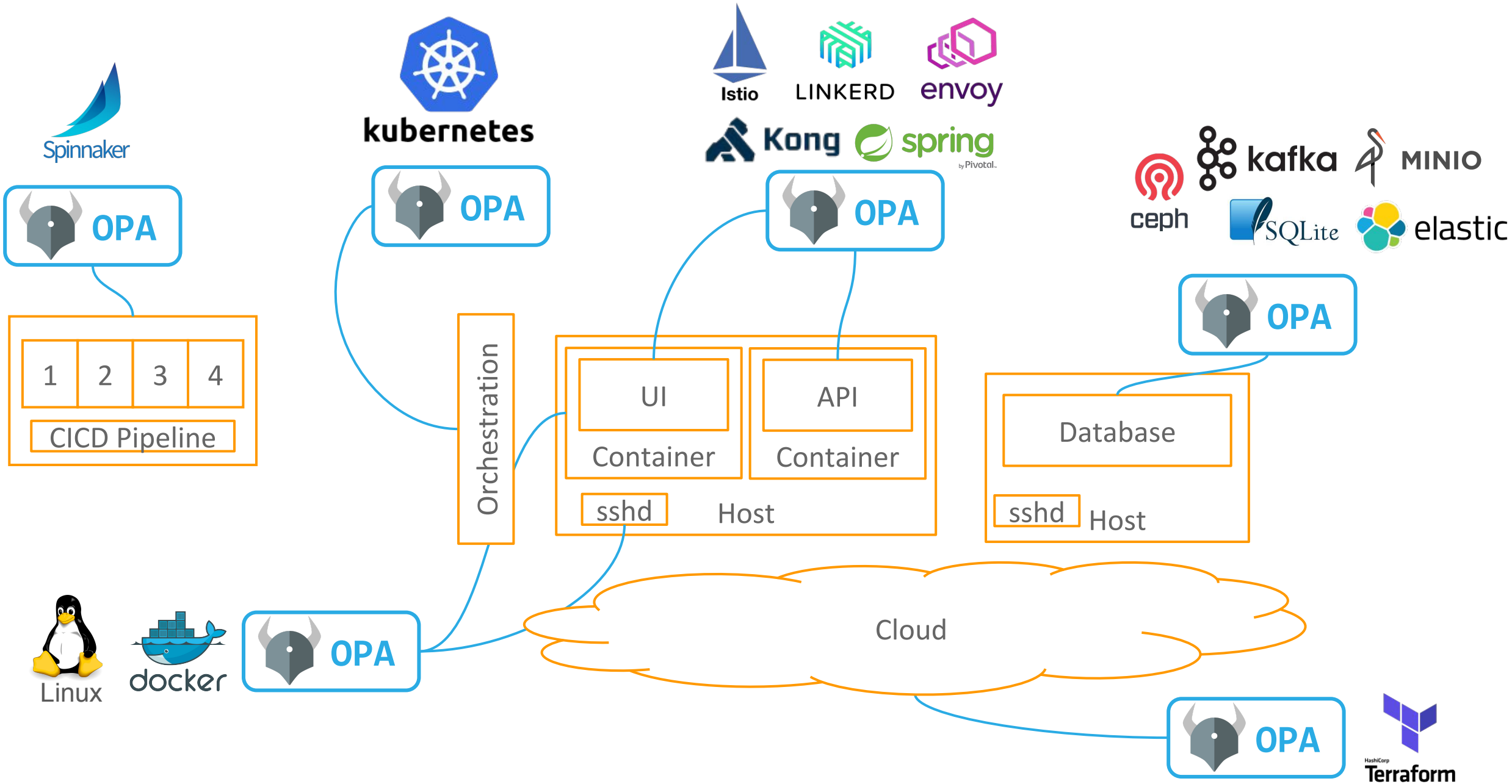
Use Case Deep Dive

New and Future Features

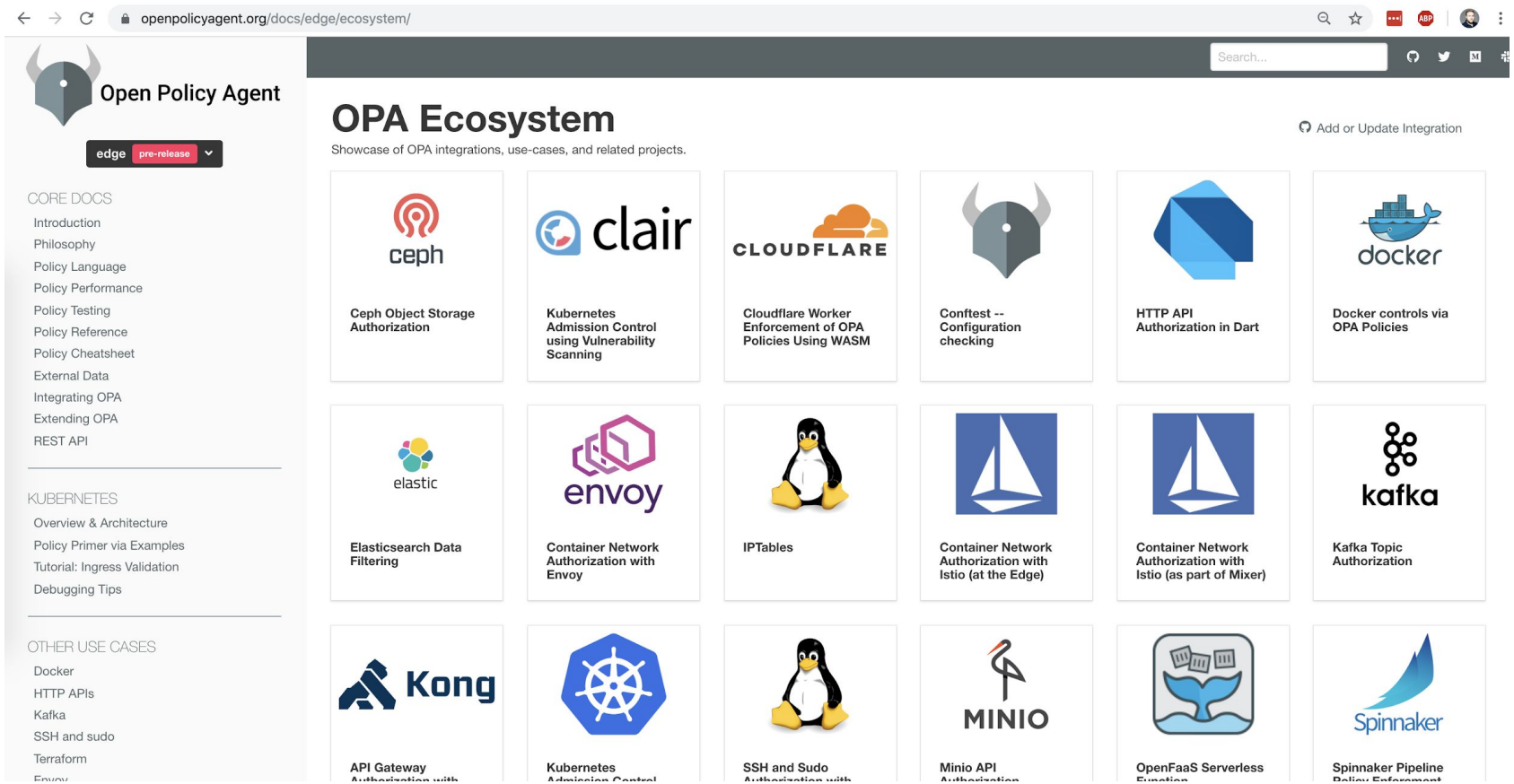
Cloud-native Diversity/Dynamism Make Policy Management Challenging



OPA: Unified Policy Across the Stack



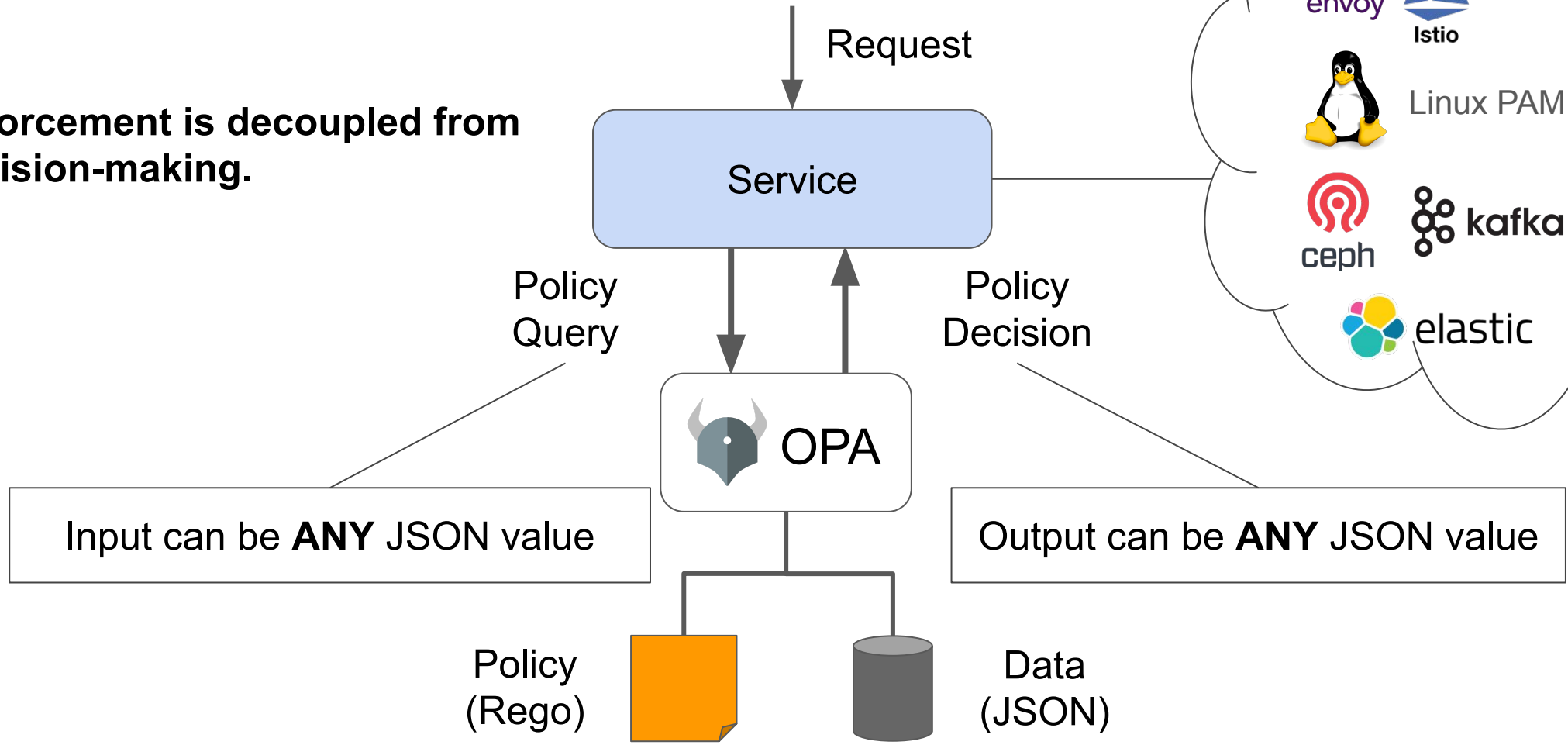
OPA Integration Index

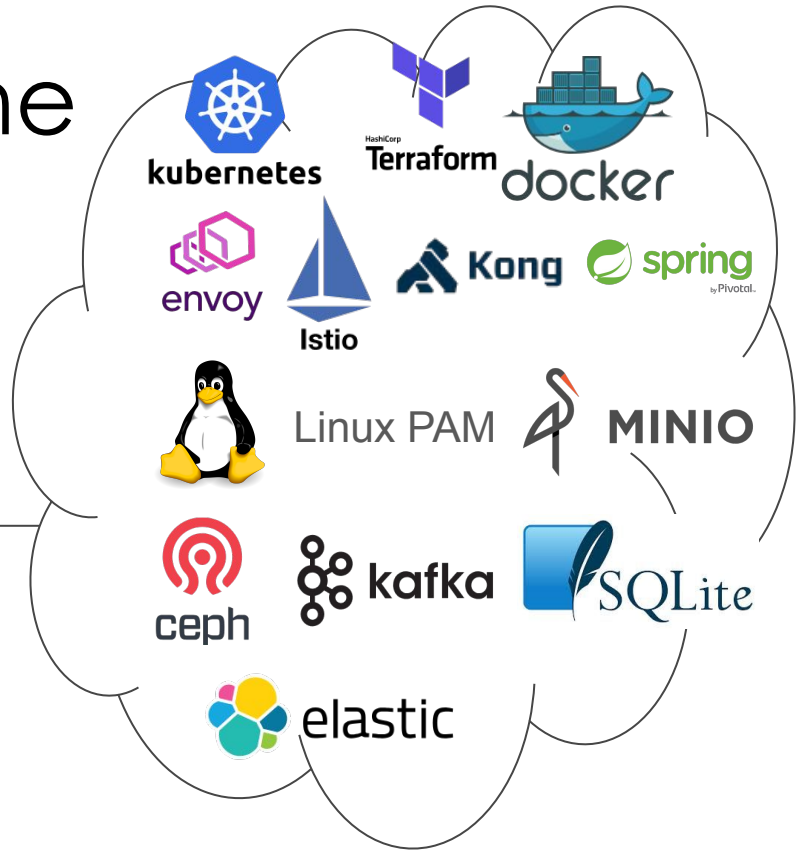


openpolicyagent.org

OPA: General-purpose Policy Engine

Enforcement is decoupled from decision-making.

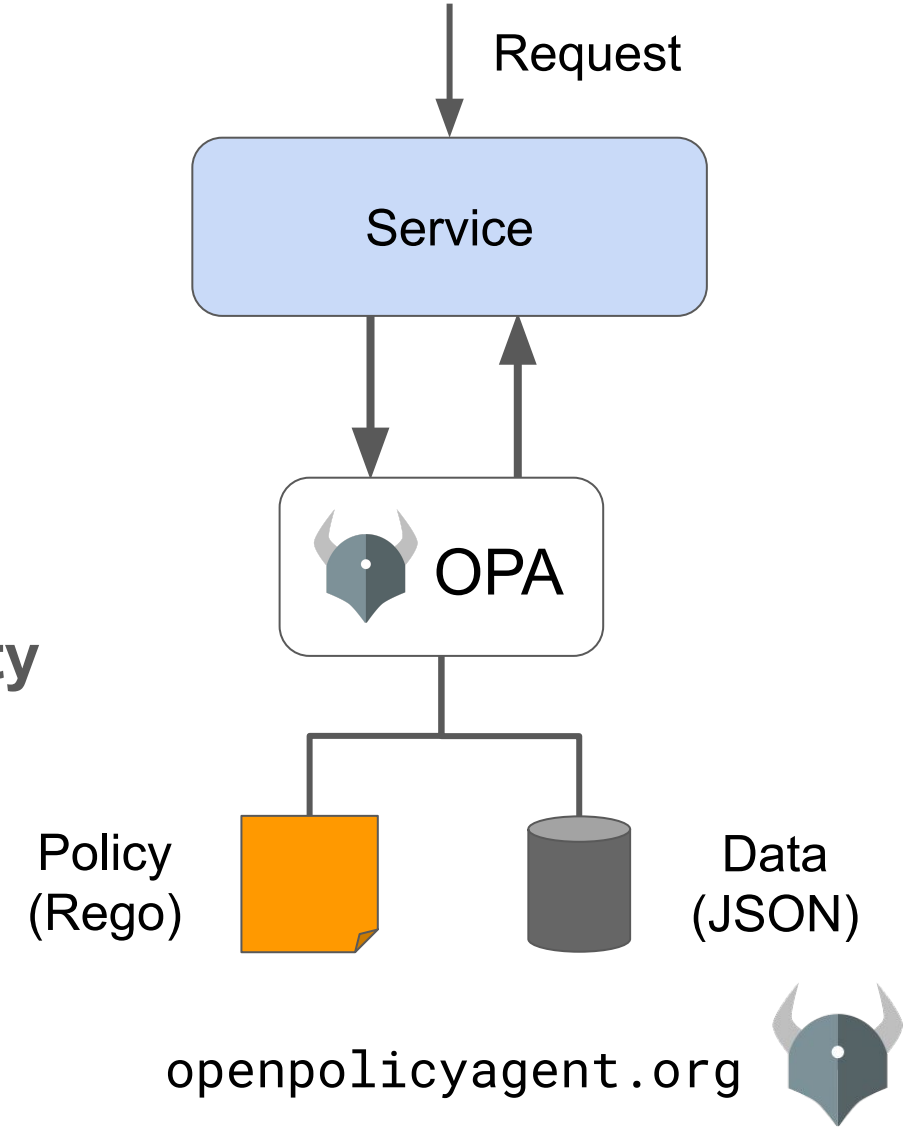




openpolicyagent.org

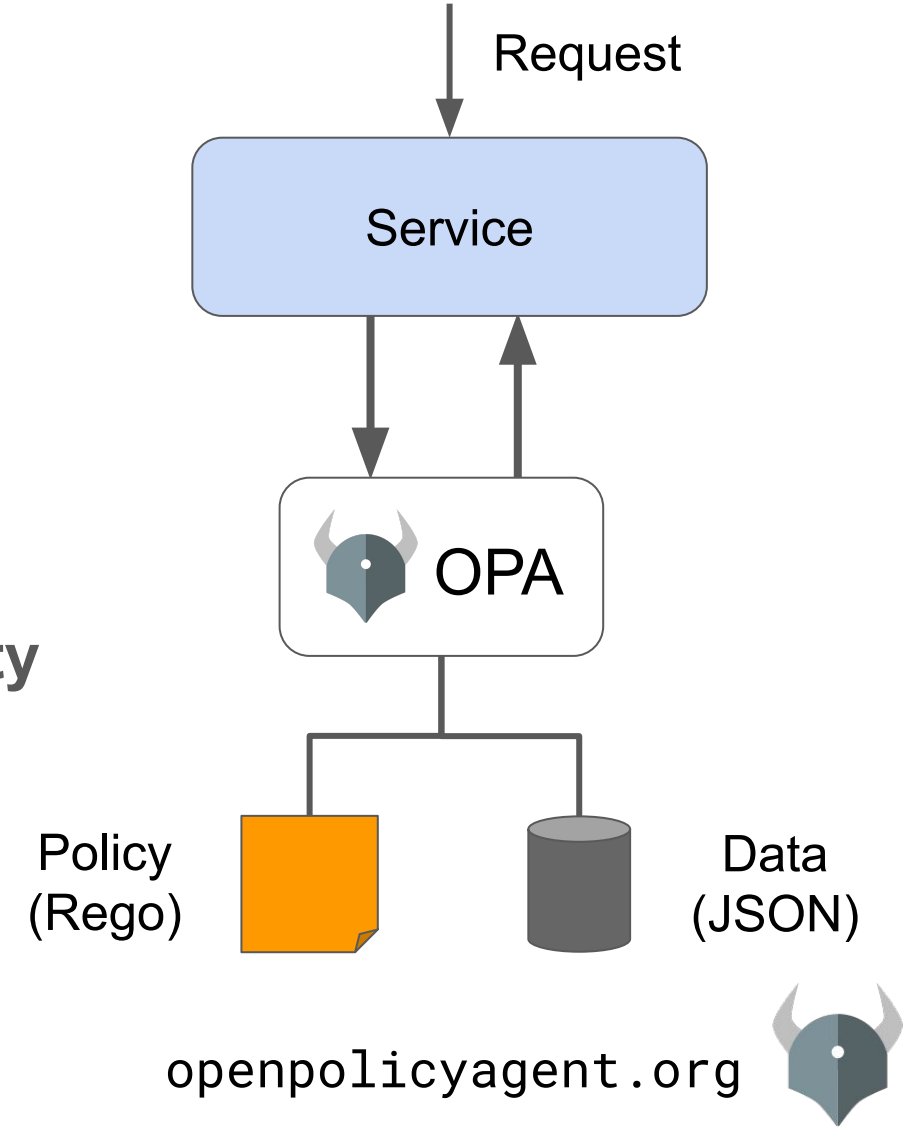
OPA: Policy-as-code

- **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.



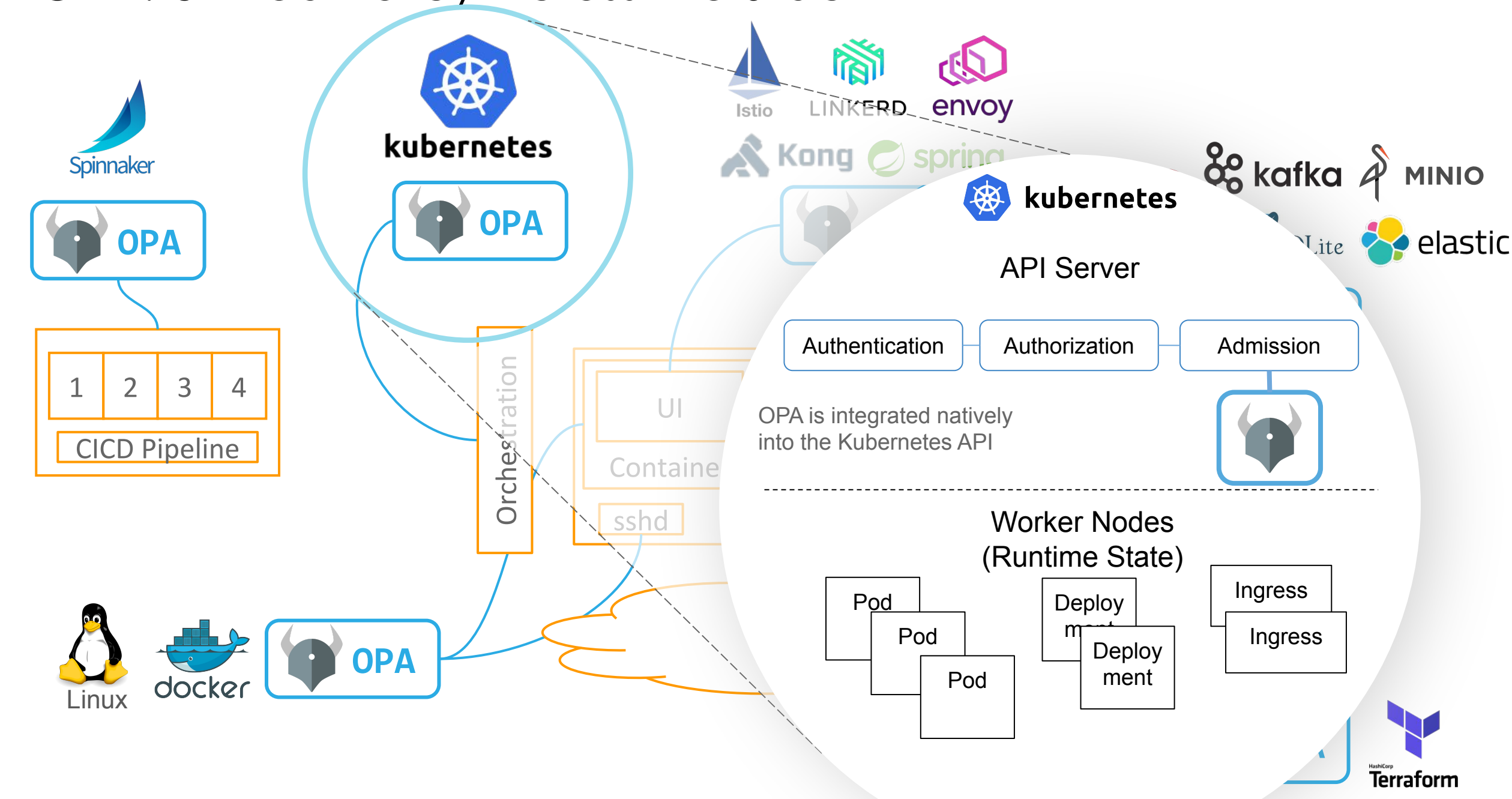
OPA: Policy-as-code

- **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.



Use Case Deep Dive

OPA: Unified Policy Across the Stack



Kubernetes Policy Example

[OPA Playground (with flat policy)]
[OPA Playground (with rich policy)]
[OPA Playground (no policy)]

JSON/YAML from Kubernetes

```
apiVersion: admission.k8s.io/v1beta1
kind: AdmissionReview
request:
  kind:
    group: ''
    kind: Pod
    version: v1
  namespace: opa
  object:
    metadata:
      labels:
        app: nginx
      name: nginx
      namespace: opa
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
        volumeMounts:
        - mountPath: "/var/run/serviceaccount"
          name: default-token-tm9v8
          readOnly: true
  operation: CREATE
```

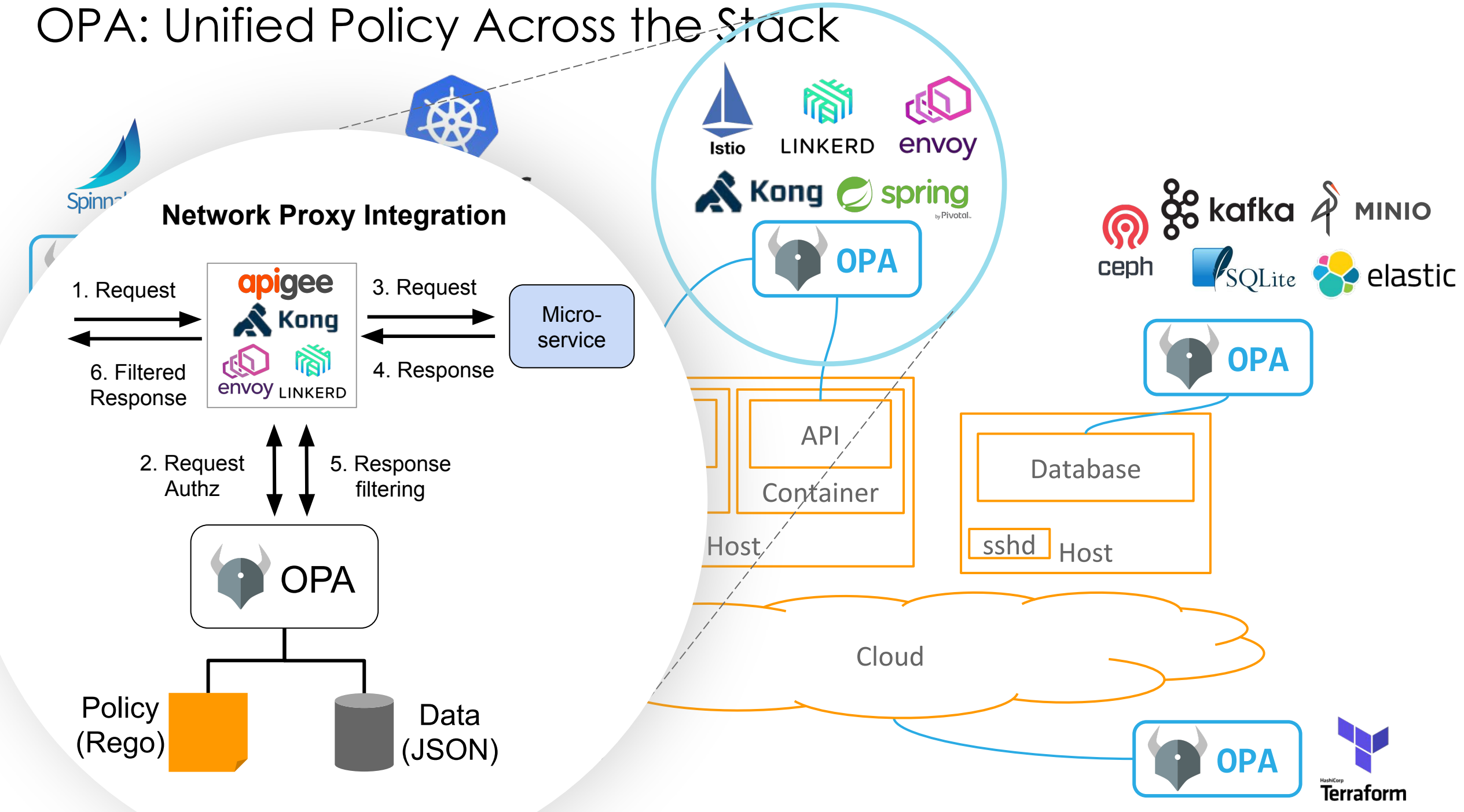
OPA Policy: All images come from a trusted registry

```
package kubernetes.admission

deny[msg] {
    input.request.kind.kind == "Pod"
    some i
    image := input.request.object.spec.containers[i].image
    not startswith(image, "hooli.com")
    msg := sprintf("image comes from bad registry: %v", [image])
}
```

openpolicyagent.org

OPA: Unified Policy Across the Stack



Envoy Policy Example

[OPA Playground (with policy)]
[OPA Playground (no policy)]

JSON/YAML from Envoy

```
parsed_path: [“api”, “v1”, “products”]
attributes:
 source:
  address:
   Address:
    SocketAddress:
     address: "172.17.0.10"
     PortSpecifier:
      PortValue: 36472
 destination:
  address:
   Address:
    SocketAddress:
     address: "172.17.0.17"
     PortSpecifier:
      PortValue: 9080
 request:
  http:
   id: 13359530607844510314
   method: GET
   headers: ...
   path: "/api/v1/products"
   host: "192.168.99.100:31380"
   protocol: "HTTP/1.1"
```

OPA Policy: Allow all GET and some PUT

```
package envoy.authz

# everyone can read everything
permit {
    input.attributes.request.http.method == "GET"
}

# writes dependent on source
permit {
    input.attributes.request.http.method == "PUT"
    input.parsed_path = ["v1", "deployment", x]
    src := input.attributes.source.address.Address.SocketAddress.address
    net.cidr_contains("172.28.0.0/16", src)
}
```

openpolicyagent.org

New and Future Features

Future: Automated Complexity Analysis

Complexity: O(input)

```
package kubernetes.admission

deny[msg] {
    input.request.kind.kind == "Pod"
    some i
    image := input.request.object.spec.containers[i].image
    not startswith(image, "hooli.com")
    msg := sprintf("image comes from bad registry: %v", [image])
}
```

Complexity: O(1)

```
package envoy.authz

# everyone can read everything
permit {
    input.attributes.request.http.method == "GET"
}

# writes dependent on source
permit {
    input.attributes.request.http.method == "PUT"
    input.parsed_path = ["v1", "deployment", x]
    src := input.attributes.source.address.Address.SocketAddress.address
    net.cidr_contains("172.28.0.0/16", src)
}
```

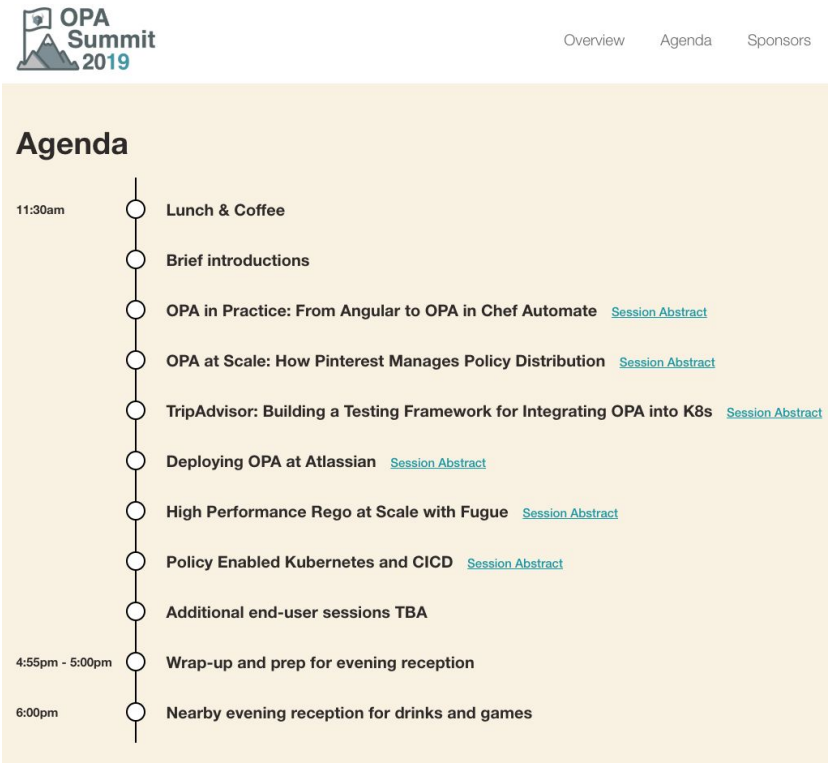
Before: Application owners, K8s admins, microservice integrators can delegate the authoring of policy to other people.

After: Admins can also dictate how rich the policies are that those other people can write.

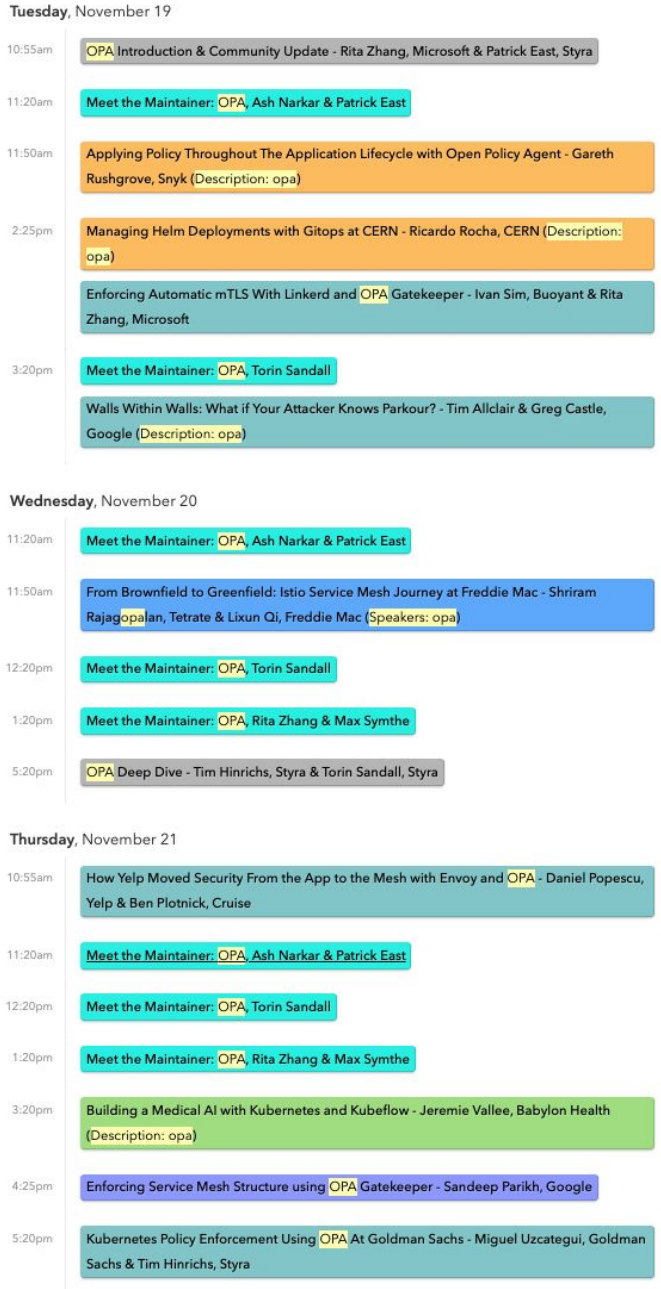
What's New?

- WebAssembly compiler
- Demo: Partial Evaluation Improvements
  - With keyword
  - Comprehensions

OPA Summit



Kubecon San Diego



Q&A

Torin Sandall

Engineer at Styra
Co-creator of OPA

@tsandall on OPA
@sometorin

Tim Hinrichs

Co-founder & CTO at Styra
Co-creator of OPA

@tim on OPA
@tlhinrichs