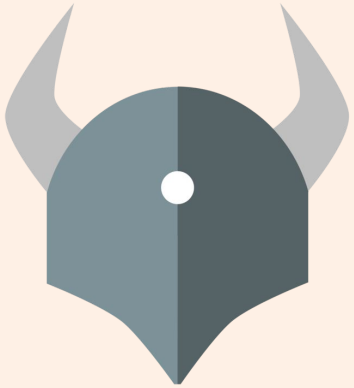
KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon Europe 2022

Open Policy Agent Introduction + Deep Dive



Anders Eknert, Styra
Will Beason, Google

Open Policy Agent Introduction + Deep Dive

KubeCon CloudNativeCon Europe 2022

- Introduction to Open Policy Agent
- Project updates
- Introduction to Gatekeeper
- Project updates

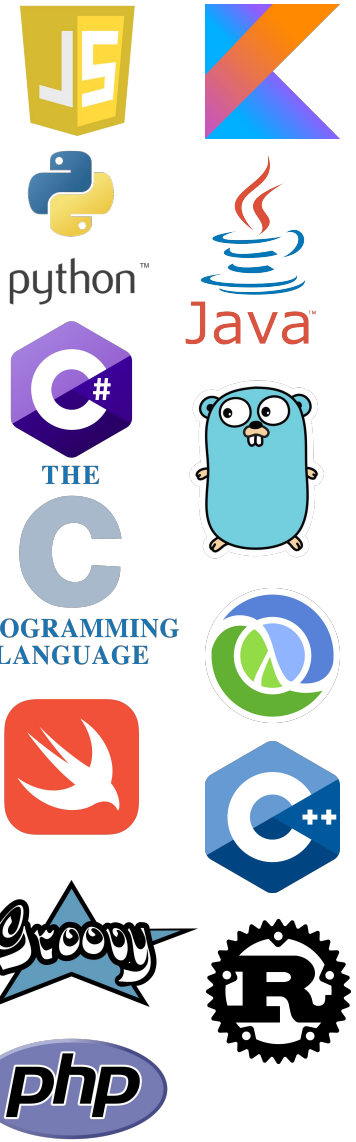
Anders Eknert
Developer Advocate
Styra

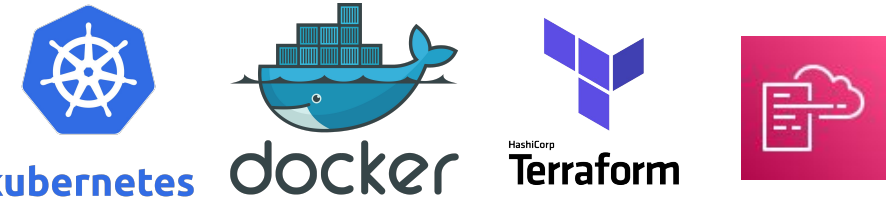
Will Beason
Software Engineer
Google

Open Policy Agent

Introduction

Open Policy Agent — Introduction

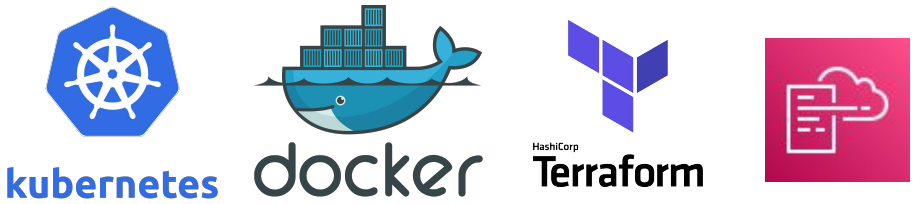




Challenge:

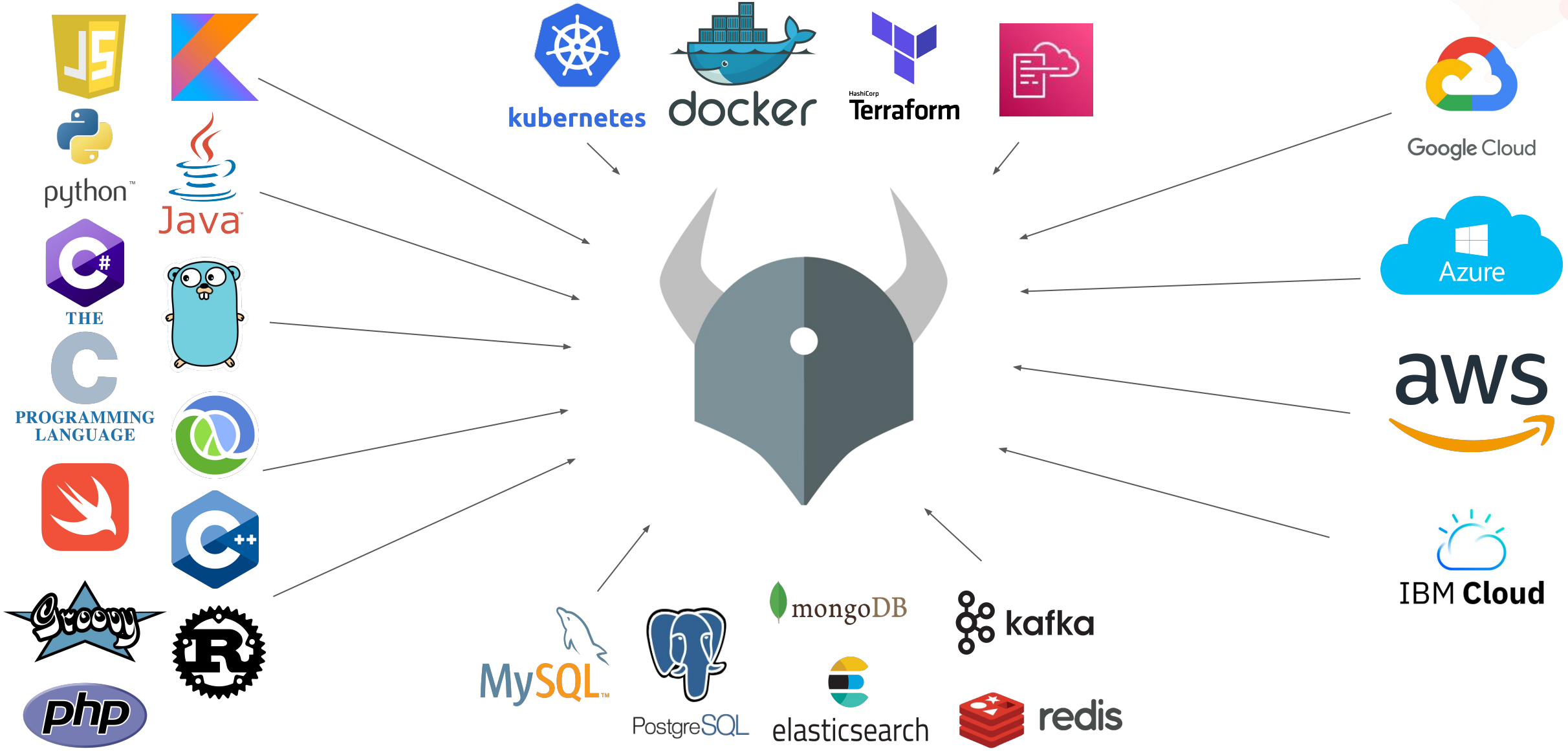
Manage policy in increasingly distributed, complex and heterogeneous systems

Open Policy Agent — Introduction



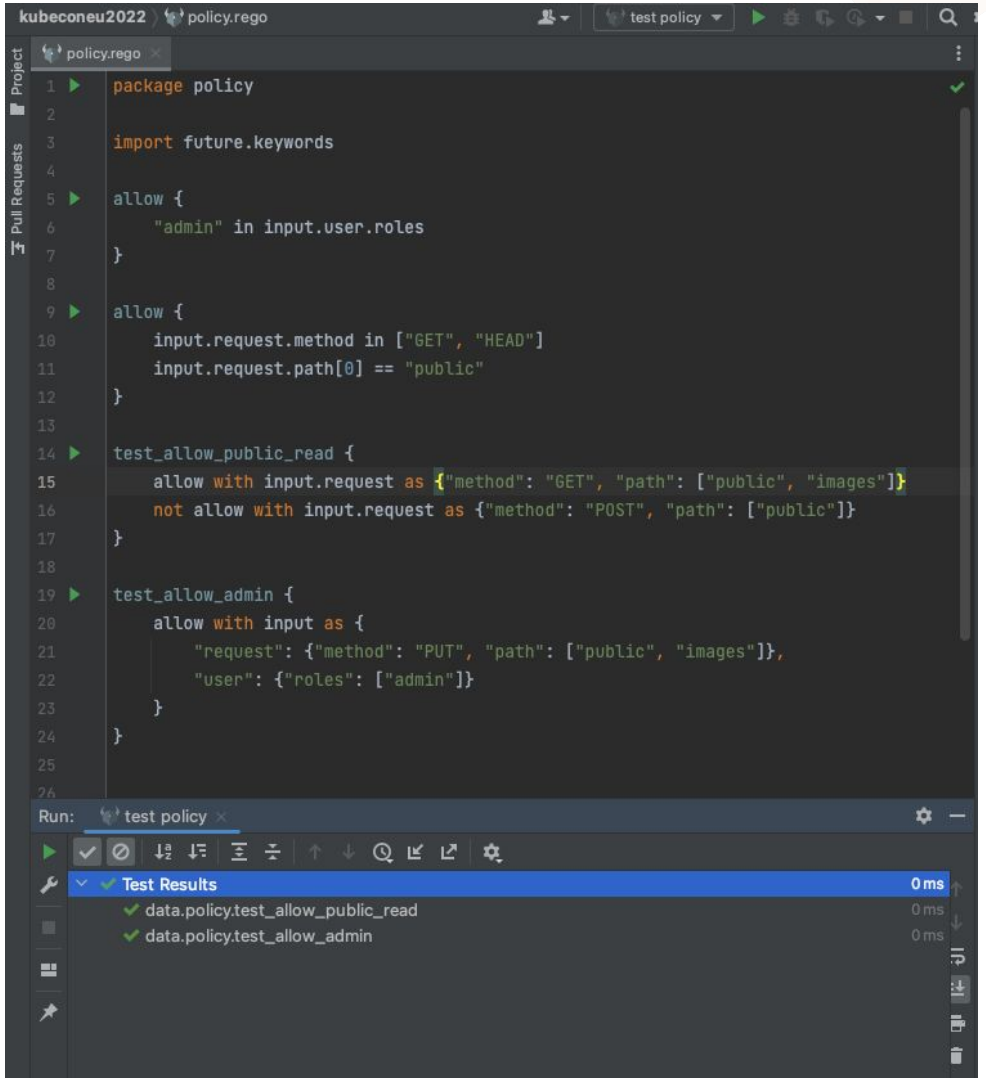
Goal:
Unify policy across the cloud native stack

Open Policy Agent — Introduction



What is policy?

- **Policy is a set of rules** — organizational rules, permissions for app authorization, Kubernetes admission control, infrastructure, builds and deployment, data filtering… and much more
- Treating **policy as code** provides all the benefits of treating *anything* as code — collaboration, peer review, testing, static analysis, linters, and much more. No more PDF documents!
- **Decoupling policy** from application and business logic means policy can change independently of application life cycle. Policy may be shared across teams and functions. Clear separation of responsibilities.

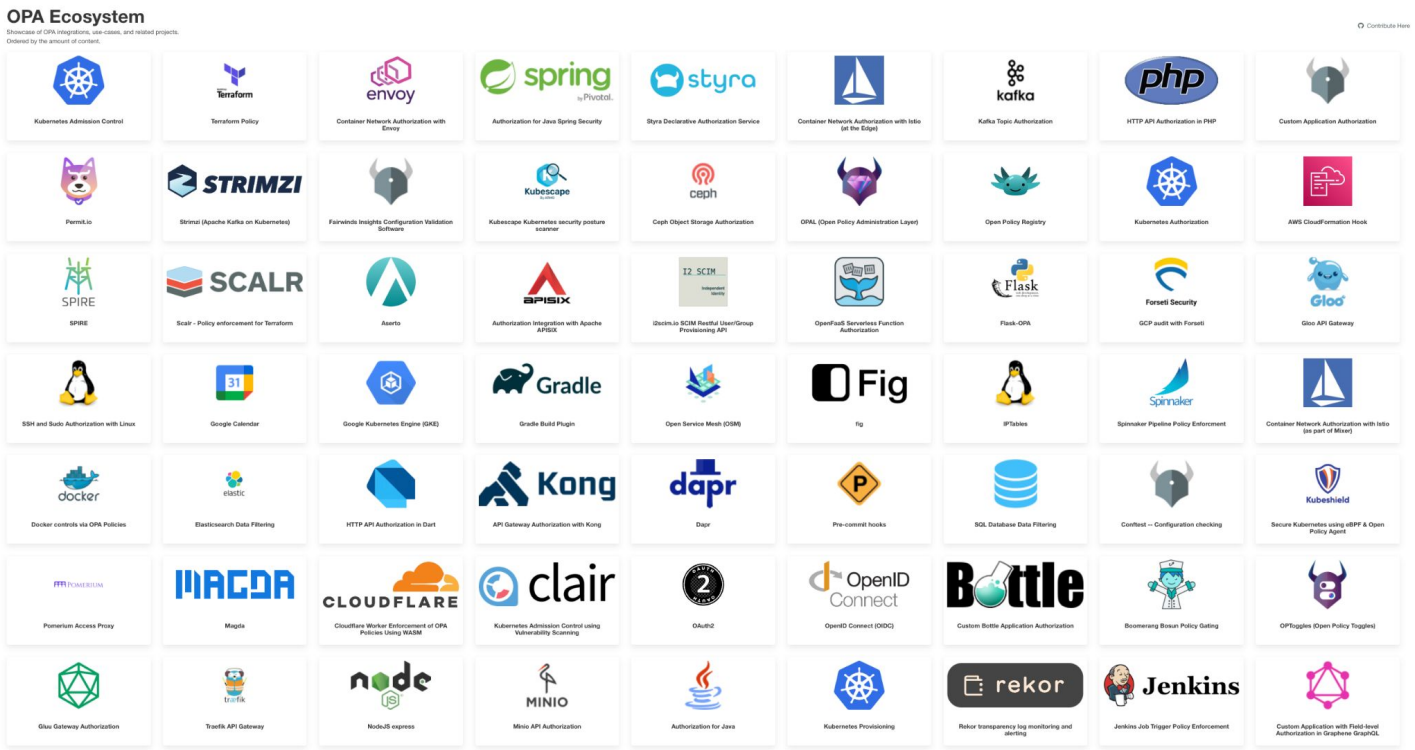


Open Policy Agent

- Open source general purpose **policy engine**
- As of February 2021, **Graduated** CNCF project
- **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**

Community in Numbers

- **250+** contributors
- **70+** integrations
- “Used by” **800+** GitHub projects
- **6600+** Github Stars
- **5800+** Slack users
- **130+** million downloads
- Ecosystem including **Conftest**, **Gatekeeper**, VS Code and IntelliJ editor plugins.



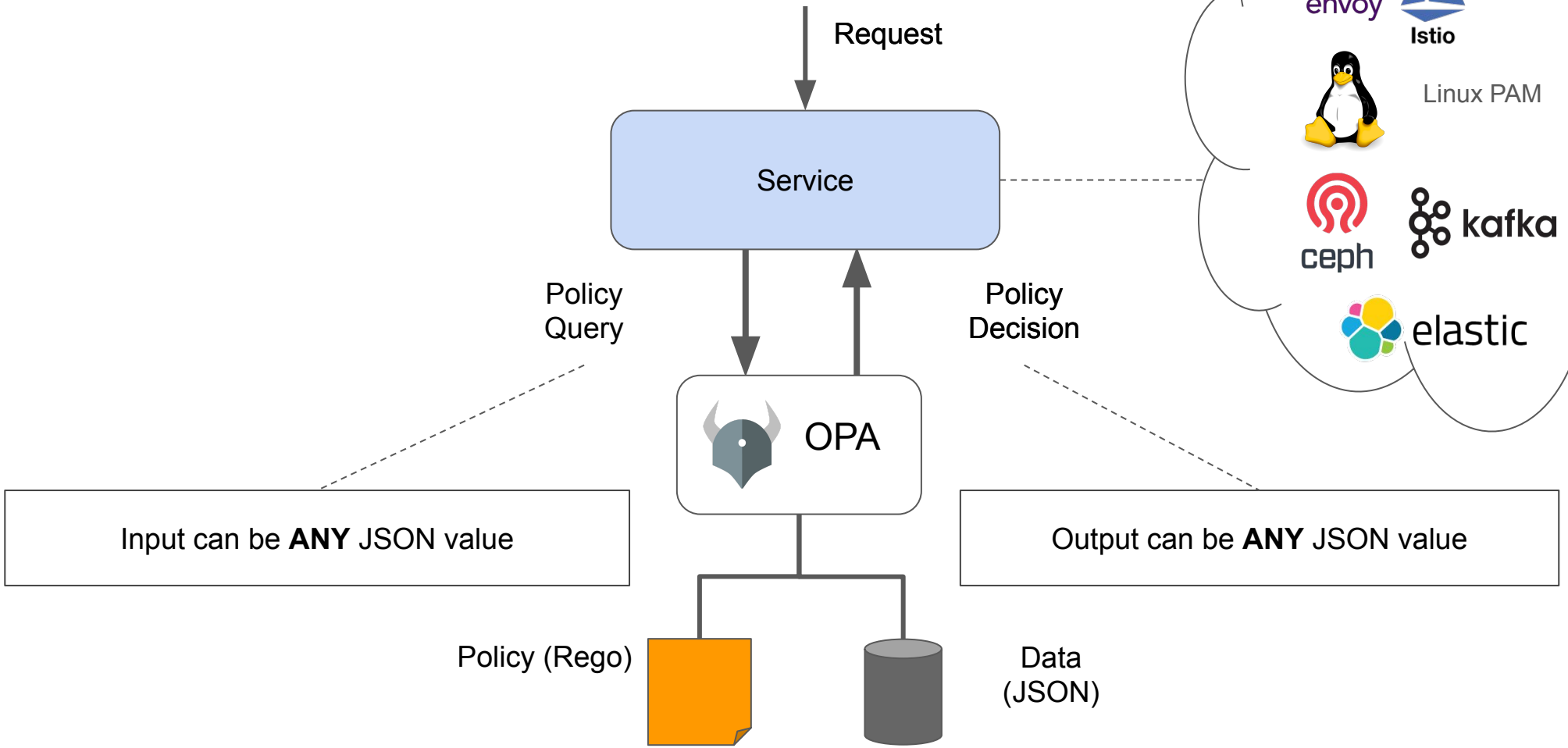
Open Policy Agent

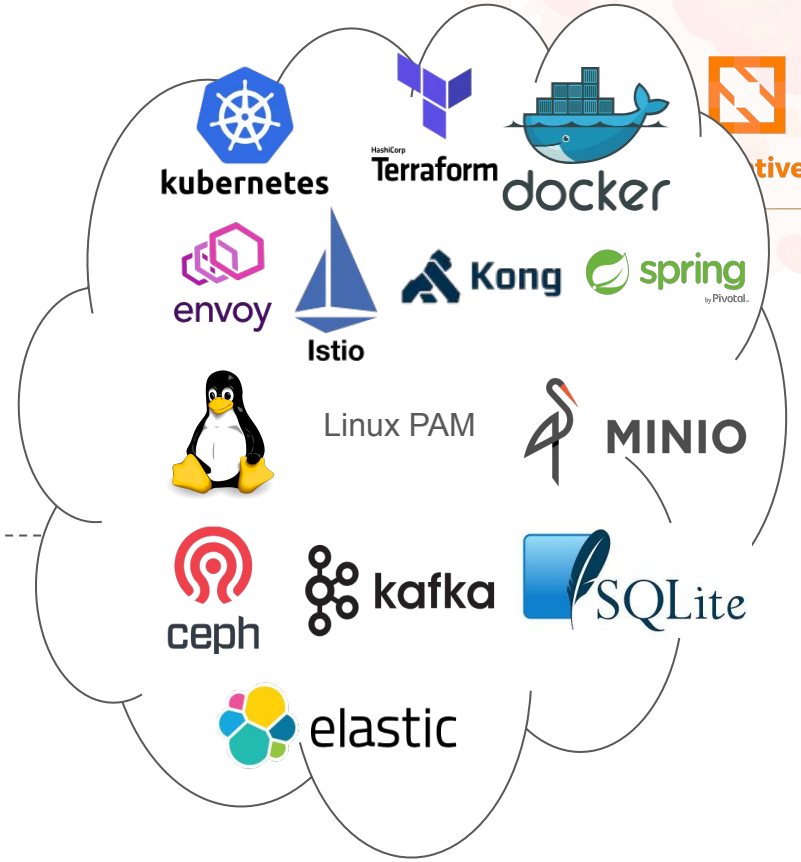
How does it work?

Policy Decision Model

Rego

Policy Decision Model





Open Policy Agent — Rego

- **Rego** — declarative high-level policy language
- Allows you to describe policy across the whole cloud-native stack
- 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
- Policy testing made easy with provided unit test framework
- Well documented! https://www.openpolicyagent.org/docs/latest/
- Try it out! https://play.openpolicyagent.org/

Teach yourself Rego in one minute


```
allow := true {
    input.request.method == "GET"
    input.request.path == ["users", input.request.user.name]
}
```

VS

```
var allow
if (input.request.method == "GET" &&
    input.request.path[0] == "users" &&
    input.request.path[1] == input.request.user.name) {
   allow = true
}
```

Open Policy Agent

Project Updates

Open Policy Agent — Project Updates

New keywords: `in`

- `in` for membership checks
  - Previously
    - `"admin" == input.user.groups[_]`
  - Now
    - `"admin" in input.user.groups`
    - `not input.request.method in {"HEAD", "GET"}`

Open Policy Agent — Project Updates

New keywords: `some … in`

- `some … in` for iterating over collections
  - Previously
    - ```
      value := input.user.groups[key]
      # do something with key/value
      ```
  - Now
    - ```
      some key, value in input.user.groups
      # do something with key/value
      ```

Open Policy Agent — Project Updates

New keywords: `every`

- `every` for expressing `FOR ALL`
  - Previously
    - ```
      internal_containers := [c | c := containers[_]; startswith(c, "acmecorp/")]
      count(internal_containers) == count(containers)
      ```
  - Now
    - ```
      every container in containers {
          startswith(container, "acmecorp/")
      }
      ```

Open Policy Agent — Project Updates

Notable new features

- Delta bundles
- Strict mode (`opa check --strict`)
- Metadata annotations

  ...and builtins for metadata introspection
- OCI bundle registry support
- Disk storage
- Function mocking

```rego
# METADATA
# title: kubernetes.pods
# description: rules for pods
# custom:
#   severity: medium
package kubernetes.pods

import future.keywords

# METADATA
# title: require security context
# description: ensure a securityContext is provided for the pod or all included container(s)
# custom:
#   severity: high
deny[{"reason": message, "severity": severity}] {
    not pod_has_security_context
    not all_containers_have_security_context

    message := "Pod does not provide securityContext, and not found in all containers"
    severity := rule_severity(rego.metadata.chain())
}

```

Open Policy Agent — Roadmap

Planned improvements

- Built-in functions to better support GraphQL
- Named built-in function arguments (for documentation and editor integrations)
- Dependency management!
- Richer test result diff reports
- Optimization flag for more tools than `opa build` (i.e. `opa eval`, `opa test` …)
- Include non-deterministic values (like `time.now_ns()`, `http.send()`, etc) in decision logs

Open Policy Agent — Project Updates

Ecosystem

- **OPA AWS CloudFormation Hook**: Policy-powered provisioning of AWS CloudFormation resources
- **setup-opa**: GitHub Action for installing OPA as part of GitHub workflows
- **SansShell**: A non-interactive daemon for host management, where any action is authorized against policy
- **Reposaur**: Audit your GitHub data using Rego policies
- **opactl**: Turn Rego rules into CLI commands

Gatekeeper + Frameworks

Updates in v3.8.x

Outline

open-policy-agent/frameworks

- Performance improvements
- Interface changes

open-policy-agent/gatekeeper

- External Data
- gator CLI
- Other changes

Gatekeeper

open-policy-agent/frameworks

Frameworks — Performance Improvements

100 Templates / 1,000 Constraints

Frameworks — Specific Improvements

- No more quadratic Template load time!
- Aside: take performance changes with a grain of salt
  - Non-frameworks/Gatekeeper performance is not measured
  - Transient vs. persistent memory
    - Gains here are (mainly) reducing transient memory usage
- 3x-20x speedup in Template compilation
  - No more quadratic load/unload times
  - Removing Templates is almost instantaneous
- 2x speedup in adding Constraints
- 2x-3x speedup in evaluating Constraints
  - 100x speedup in running match criteria
  - Use-case dependent
- ~20x reduction in persistent Audit memory usage
  - 100 Templates/1,000 Constraints
  - 200 Namespaces + 200 ConfigMaps
  - ~20GB -> 1GB

Frameworks — Behavioral Changes

- OPA v0.39.0
  - "future" keywords
- Client/Driver
  - Interfaces reworked
  - No longer cache Constraint status
  - No more Audit
    - Callers must implement it themselves
- TargetHandler
  - Interface reworked
    - Defines match criteria in Go
  - Templates sandboxed from each other
    - Still point to the same Rego storage
  - Targets may cache referenced data
    - No changes to sync requirements for Gatekeeper

Gatekeeper

open-policy-agent/gatekeeper

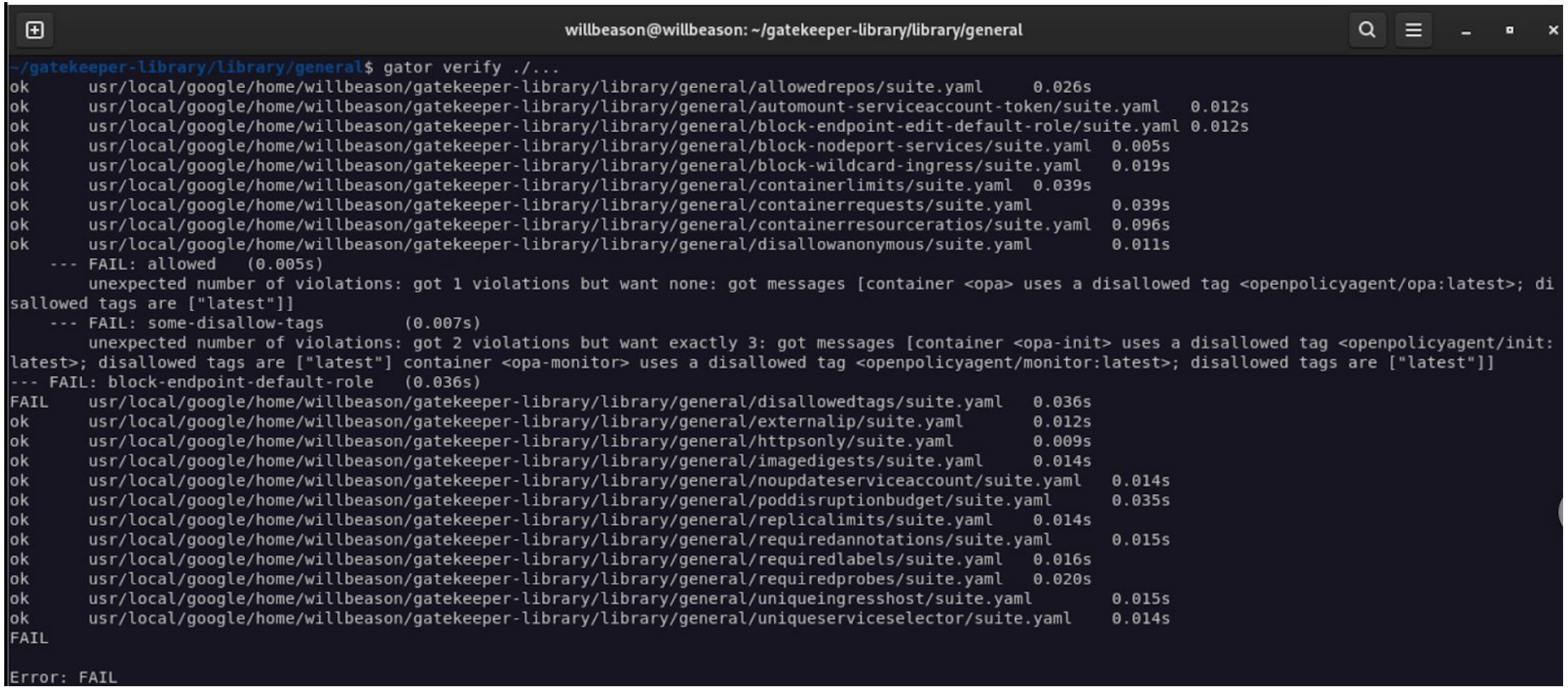
Gatekeeper — New Feature: External Data

- Communicate with external systems
- More secure than http.send
- Can batch requests natively
- Improved cached hits
- Designed protocol to be use-case neutral
  - Assume key-value store
  - Configurable TTL
- Use cases
  - LDAP: Lets users integrate.
    - Can limit who can change specific fields.
    - Auto-label resource with team metadata.
  - Container Vulnerability
    - Check for CVE vulnerabilities against a database of scanned images
    - Only deploy trusted workloads/secure builds
- Mutation
  - Latency constrained
  - Parallel, synchronous evaluation
  - Can only use on string-valued data

Gatekeeper — New Feature: gator CLI

- gator verify
  - Similar to "conftest verify"
  - Unit tests for Templates and Constraints
  - Can test the contents of the output
- gator test
  - Similar to "conftest test"
  - Validate all resources in a repository against Constraints in that repository

Gatekeeper — New Feature: gator CLI



Gatekeeper — Other Improvements

- Prometheus metrics for conflicting mutators
- Helm: Can configure to remove webhooks before uninstalling Gatekeeper
  - Don't take your cluster out while uninstalling!

Thank you!

Questions?