KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

Open Policy Agent (OPA): Intro & Deep Dive

Sebastian Spaink & Johan Fylling, Apple





Sebastian Spaink

OPA maintainer

Contributing since 2023

`github.com/sspaink`



Johan Fylling

OPA maintainer

Contributing since 2021

`github.com/johanfylling`

KubeCon CloudNativeCon
North America 2026



Introduction to OPA



Policy is scattered

Hardcoded in services, webhooks, CI and AI agents: **different languages, different semantics.**

Web service (Go)



```go
if user.Role != "admin" &&
   r.Method != "GET" {
    w.WriteHeader(403)
}
```

Admission webhook (Python)



```python
for c in pod["spec"]["containers"]:
    if c["image"].endswith(":latest"):
        return deny("pin image tags")
```

CI script (Bash)



```bash
if [[ "$BRANCH" == "main" &&
      "$APPROVALS" -lt 2 ]]; then
  exit 1
fi
```

AI agent tool call (TypeScript)



```typescript
if (call.tool === "shell" &&
    call.args.cmd.includes("rm -rf")) {
  return deny("destructive");
}
```

One Rego policy for all four



One language, one policy for every case: **test, lint & review it like any code.**

Web service

```
api.allow if input.user.role == "admin"

api.allow if input.method == "GET"
```

Admission webhook

```
k8s.deny contains "pin image tags" if {
    some c in input.spec.containers
    endswith(c.image, ":latest")
}
```

CI script

```
ci.deny contains "needs 2 approvals" if {
    input.branch == "main"
    input.approvals < 2
}
```

AI agent tool call

```
tools.deny contains "destructive" if {
    input.tool == "shell"
    contains(input.args.cmd, "rm -rf")
}
```

A declarative language

Admission webhook (Python)



```python
for c in pod["spec"]["containers"]:
    if c["image"].endswith(":latest"):
        return deny("pin image tags")
```

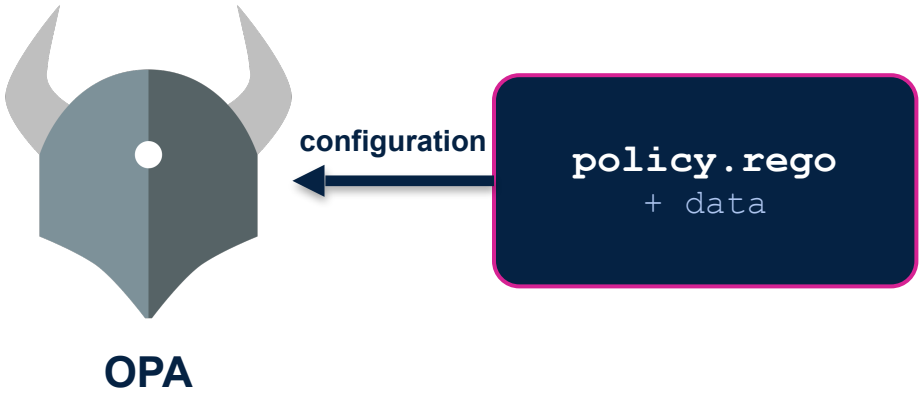
Admission webhook (Rego)



```
k8s.deny contains "pin image tags" if {
    some c in input.spec.containers
    endswith(c.image, ":latest")
}
```

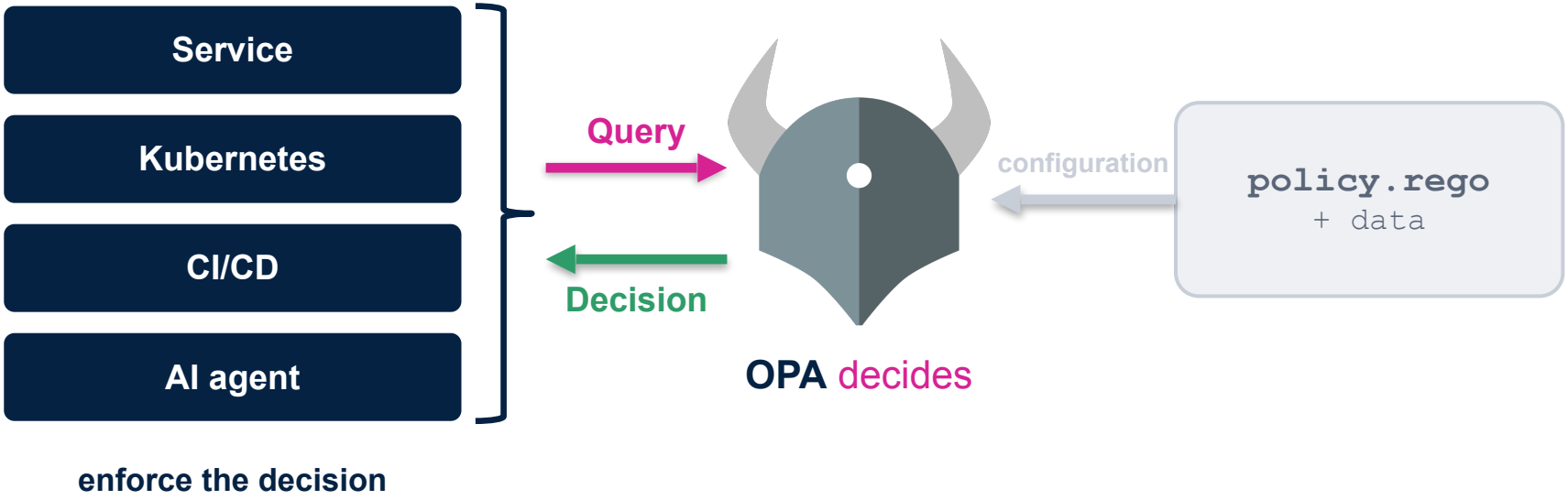
OPA loads your policy and data

Policy and data are **configuration, managed separately from your code.**



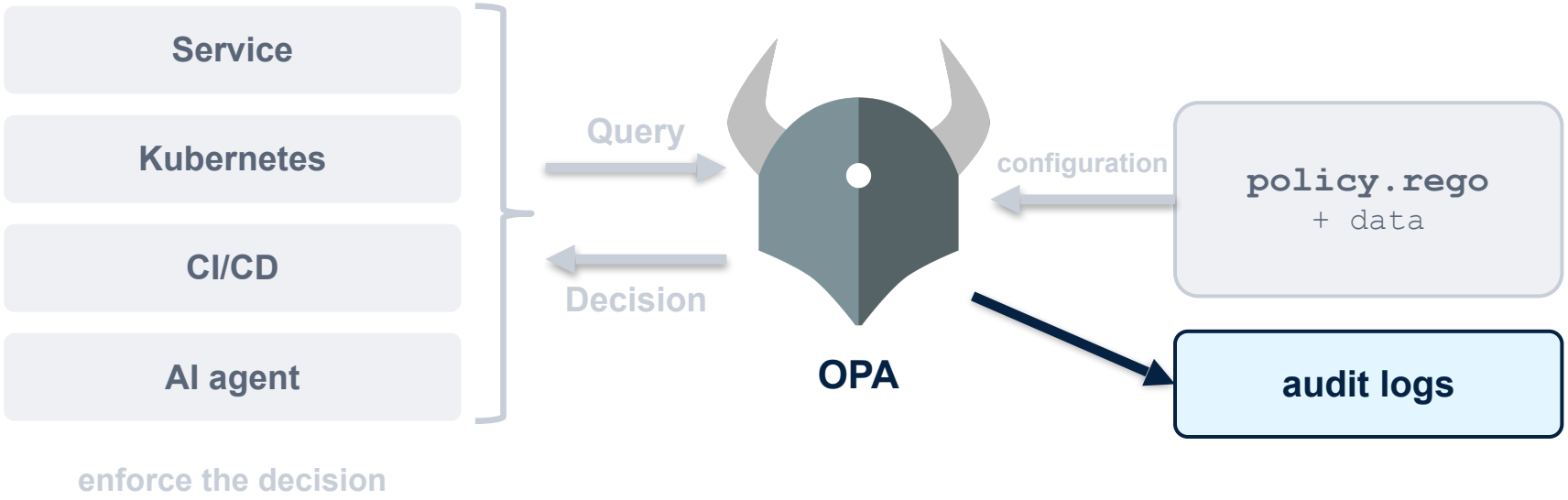
Decouple policy decisions from enforcement

Your systems **ask**, OPA **decides**, your systems enforce.

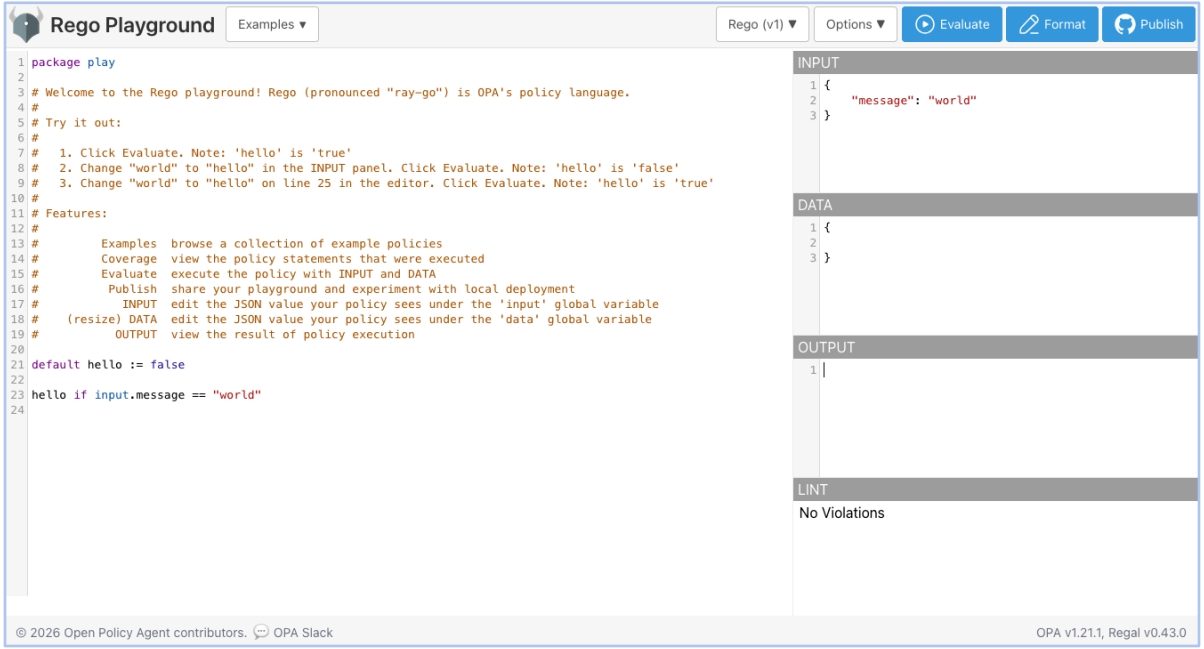


Every decision is logged

Decision logs: an audit trail of what was allowed or denied, and why.



Try it in 60 seconds



play.openpolicyagent.org

Seven releases since KubeCon EU



KubeCon CloudNativeCon
North America 2026



Rego language updates



Expressing `and` & `or` with Rego

```
allow if {
    input.method == "GET"
    admin_or_public
}

admin_or_public if input.user.admin

admin_or_public if input.resource.public
```

and

or

Rego gets `and` & `or`

v1.20: combine conditions inside one rule body, no helper rules needed.

Before

```
allow if {
    input.method == "GET"
    admin_or_public
}

admin_or_public if input.user.admin

admin_or_public if input.resource.public
```

Now

```
import future.keywords.or

allow if {
    input.method == "GET"
    input.user.admin or input.resource.public
}
```

Combining conditions with `and` & `or`

```
import future.keywords.and
import future.keywords.or

allow if input.method == "GET" and (input.user.admin or input.resource.public)
```

Rego gets safer negation (`not`) semantics

v1.17: an undefined value inside `not` no longer skips the rule.

policy.rego

```
package app

deny if not endswith(input.email, "@example.com")
```

What's wrong with this policy? Is it denied?

It should deny any email that isn't `@example.com`.

policy.rego

```
package app

deny if not endswith(input.email, "@example.com")
```

A request with email

```
{"email": "john@foo.com"}
```



OPA response

```
{"deny": true}
```

A request with no email

```
{}
```



OPA response

```
{}
```

How the compiler rewrites it

```
not endswith(input.email, "@example.com")
```

Before: hoisted out of not

```
deny if {
  __local0__ = input.email
  not endswith(__local0__, "@example.com")
}
```

The lookup runs first.
With no email it fails, and the rule stops before not runs.

With the import: kept inside not

```
deny if {
  not {
    __local0__ = input.email
    endswith(__local0__, "@example.com")
  }
}
```

The lookup is part of the not.
With no email the body fails, so not succeeds.

Safer negation semantics

Before

```
package app

deny if not endswith(input.email, "@example.com")
```

Fixed

```
package app

import future.keywords.not

deny if not endswith(input.email, "@example.com")
```

deny is undefined:
the request is allowed

deny is true:
the request is denied

`input.email` is missing, so the condition is undefined. Inside `not`, undefined now counts as not holding.

One import for Rego v2

v1.22: `rego.v2` turns on and, or and the new not together.

Before: one import per keyword

```
import future.keywords.and
import future.keywords.not
import future.keywords.or
```

v1.22

```
import rego.v2
```

Same idea as `rego.v1` before OPA 1.0: **opt in today, ready for v2.**

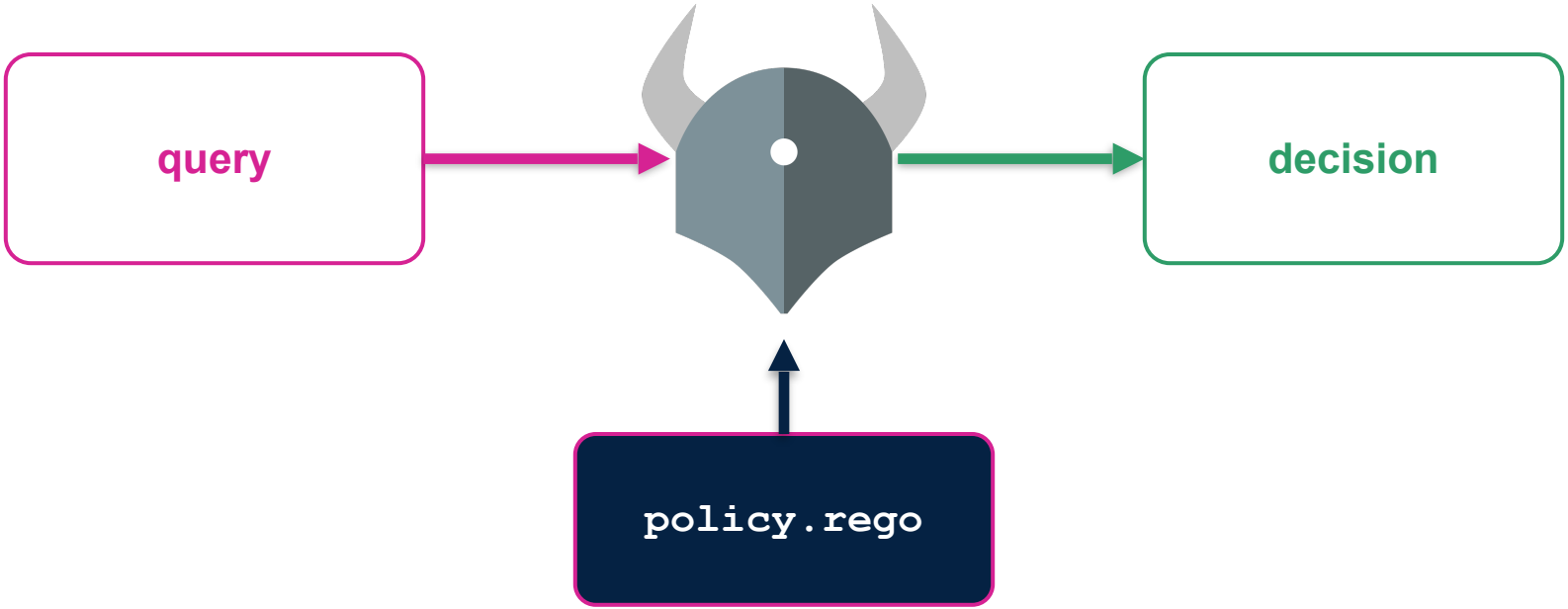
KubeCon CloudNativeCon
North America 2026



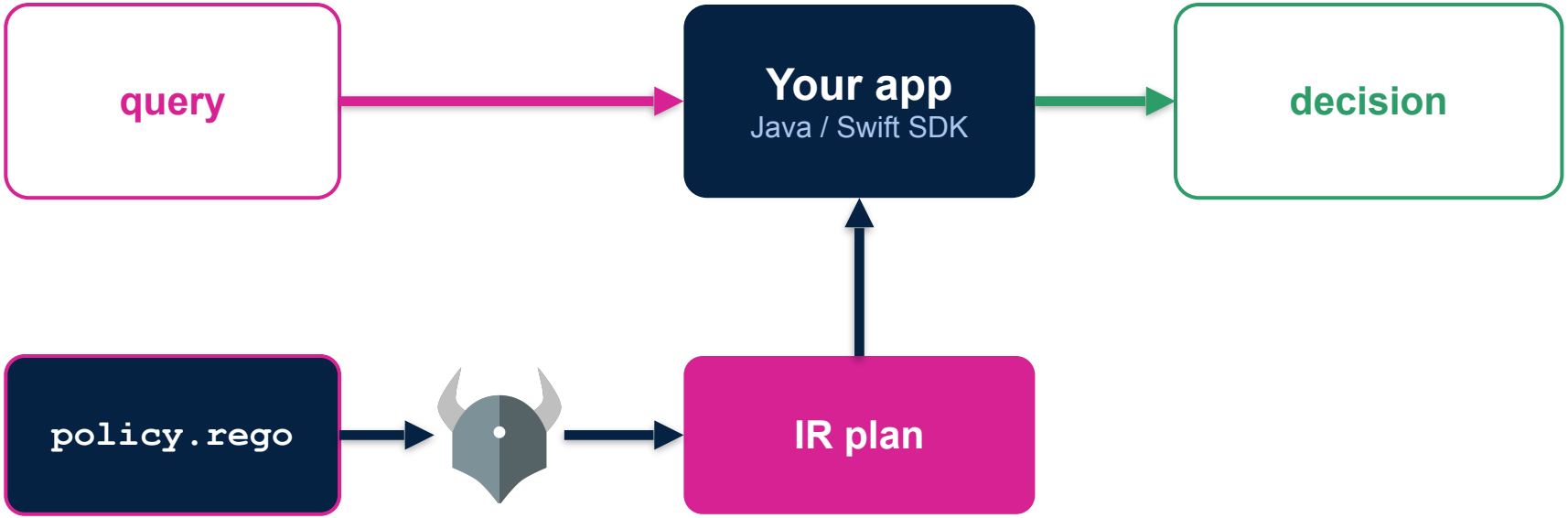
Run Rego anywhere



How does evaluation work?



How do the SDKs evaluate Rego?



Native SDKs: Rego in Java & Swift

Compile Rego once, **evaluate it in-process in Java and Swift.**

Java



A full OPA runtime for the JVM

Bundles, decision logs, status and discovery, with no OPA server to run.

github.com/open-policy-agent/java-opa-sdk

Swift



OPA for Swift apps

Evaluate policy in-process, and add swift-opa-sdk for bundle loading, logging and metrics.

github.com/open-policy-agent/swift-opa

KubeCon CloudNativeCon
North America 2026



Other highlights



Faster to start up

32%

faster compiler

Large Rego codebase:
75 → 51 ms

−44% allocations (v1.21)

39% faster data loading

12 MB JSON: 349 → 212 ms (v1.21)

29% less memory

Loading 12 MB of JSON data: 517 → 366 MiB (v1.21)

Label your rules, enrich your decision logs

v1.17: in decision logs. **v1.21:** in API responses with `?rule_labels`

Policy

```
# METADATA
# scope: package
# labels:
#   service: authz
package myapp

# METADATA
# labels:
#   reason: admin
allow if input.role == "admin"

# METADATA
# labels:
#   reason: owner
allow if input.user == input.owner
```

Decision log entry

```
{
  "input": {
    "user": "bob",
    "owner": "bob"
  },
  "result": true,
  "rule_labels": [{
    "reason": "owner",
    "service": "authz"
  }]
}
```

Config validation, now in Rego

v1.19: OPA's built-in checks are embedded Rego policies, **and typos no longer slip through.**

config.yaml

```
decision_logs:
  console: true
  reportng:    # typo
    min_delay_seconds: 5
```



Warning at startup

```
unknown configuration option
"decision_logs.reportng"
encountered
```

KubeCon CloudNativeCon
North America 2026



Community updates



Gatekeeper updates

Slide by
Jaydip Gabani



- **Operation-specific mutations**
  - Select CREATE or UPDATE using `applyTo.operations` to avoid unwanted changes during updates.
  - Existing CREATE + UPDATE behavior remains the default.
- **Richer policy context for Rego and CEL**
  - Policies can access namespace objects, including labels and annotations.
  - Expanded/generated resources retain the original admission operation.
- **Remote-cluster enforcement — alpha**
  - Run Gatekeeper in a management cluster while enforcing policies in a target cluster.
  - Status resources remain alongside Gatekeeper in the management cluster.
- **Gatekeeper-managed VAP improvements — beta**
  - Added DELETE validation and improved generated-policy stability.
  - Fewer repeated updates and corrected failure-policy defaults.
- **Better observability and operational reliability**
  - `semantic: true` identifies policy-related logs for easier filtering and monitoring.
  - Restored webhook TLS readiness and enforced ExternalData provider timeouts.

Upcoming for Gatekeeper

Slide by
Jaydip Gabani



- **Admission violation export — alpha**
  - Capture violations from denied requests that periodic audit cannot see.
  - Opt-in, disk-only, and best-effort delivery; exported records can be lost.
- **API-server audit annotations — alpha**
  - Include Gatekeeper webhook and generated VAP policy failures in Kubernetes audit logs.
  - Opt-in; requires API-server audit logging at Metadata level or higher.
- **Performance improvements**
  - Lower mutation, audit and controller overhead: faster matching, fewer allocations and updates.
  - Performance gains vary by workload.

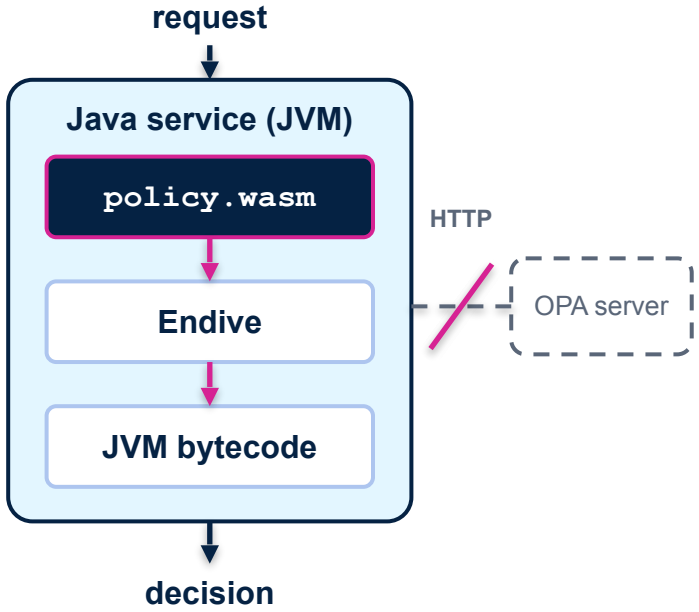
In-process OPA for Java

Slide by
Andrea Peruffo



New **opa-java-wasm 2.0.0** joined the OPA org

1. **A separate OPA server has a cost**: more infrastructure, plus a network hop per decision
2. **OPA natively emits Wasm**: `opa build -t wasm` compiles Rego to WebAssembly
3. **Endive runs it on the JVM**: a pure-Java Wasm runtime: no JNI, no native code
4. **Policies evaluate in-process**: fast, sandboxed, just a Maven dependency



`org.openpolicyagent:opa-java-wasm`

opa-java-wasm

endive

Why use Conftest?

Slide by
James Alseth



Conftest is an easy-to-use CLI for testing configurations

- All common configuration formats supported (PRs to add more are always welcome!)
- Built-in Rego functions for `parse_config_file` make it easy to write unit tests
- Integration with common test systems (GitHub Actions, JUnit, etc)

What’s new in Conftest?

Slide by
James Alseth



- **New Parsers**
  - GitHub CODEOWNERS
  - Groovy / Jenkinsfile
  - Nginx
- **Core features**
  - Source location indicators. Point to the exact line causing the violation.
  - Signed releases. You can verify the builds come from trusted conftest CI.

Conftest: coming soon

Slide by
James Alseth



We're working toward V1. ETA EOY.

- **Stable API**
- **Strict semver adherence**
  - Any breaking change is a major version bump, including breaking changes to parsers/ outputters
- **N-1 release support for bug and security fixes**
- **Better reporting of success cases – not just what failed the checks**

Other exciting projects

OPA Control Plane (OCP)
Manage policies for your OPA deployments
github.com/open-policy-agent/opa-control-plane

OPA Envoy Plugin
Enforce OPA policies with Envoy
github.com/open-policy-agent/opa-envoy-plugin

Regal
Linter and language server for Rego
github.com/open-policy-agent/regal

Java OPA SDK
Evaluate IR plans compiled from Rego in Java
github.com/open-policy-agent/java-opa-sdk

swift-opa
Evaluate IR plans compiled from Rego in Swift
github.com/open-policy-agent/swift-opa

swift-opa-sdk
Bundle loading, logging and metrics for swift-opa
github.com/open-policy-agent/swift-opa-sdk



KubeCon | CloudNativeCon

North America 2026

Get involved

Slack **slack.openpolicyagent.org**
GitHub **github.com/open-policy-agent**
Playground **play.openpolicyagent.org**

Find us after the session for questions!

