

Introduction to Open Policy Agent



Wojciech Barczyński, VP of Engineering
wojciechb@spacelift.io

Problem

Chasing the rabbit:

- Policies
- Configuration
- and ofc secure



Potential solutions

- Endless forms
- Markup language

Potential solutions

- Scripting language embedded

Policy as-a-code

- Dedicated language with safety guarantees
- Work on loosely structured data
- have a JSON support

Open Policy Agent



Created by

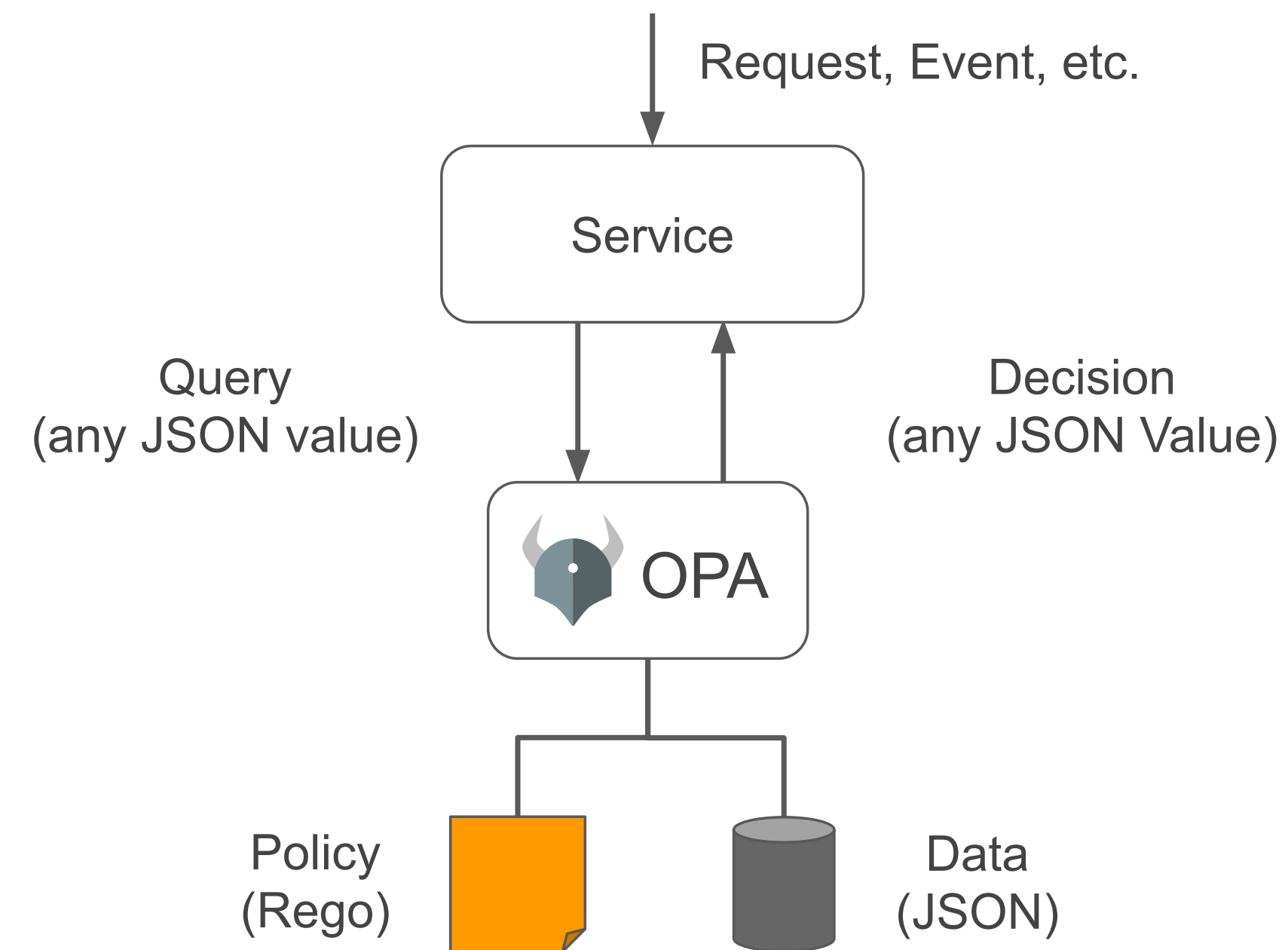


openpolicyagent.org



Open Policy Agent

1. Query/transformation language (rego)
2. Works on loosely structured data
3. With JSON and graph support



Demo

play.openpolicyagent.org

Demo

Input:

```
{  
  "session": {  
    "login": "natalia@example.com",  
    "teams": ["DevOps"],  
    "member": true  
  }  
}
```

Demo

Policy:

```
package talk_opa
teams := input.session.teams

admin { teams[_] == "DevOps" }
allow { teams[_] == "Engineering" }
deny  { not input.session.member }
```

Rego

- Declarative;
- Supports arbitrarily nested documents;
- Readable and easier for non-programmers*;

Rego

- You can run the custom code safe:
 - non Turing-complete (no loops or conditionals);
 - guaranteed to terminate;
 - ensures that queries are correct and unambiguous.
- Build-in functions

Rego



















- inspired by [Datalog](#)
- support reasoning
- extended with support for structured documents
- Rego queries are assertions on data stored in OPA

Demo

Data - global values:

```
{  
  "maintenance": false,  
  "region": "eu-west-1"  
}
```

Ecosystem

 <p>Kubernetes Admission Control</p>	 <p>Terraform Policy</p>	 <p>Container Network Authorization with Envoy</p>	 <p>Authorization for Java Spring Security</p>	 <p>Styra Declarative Authorization Service</p>	 <p>Container Network Authorization with Istio (at the Edge)</p>
 <p>Strimzi (Apache Kafka on Kubernetes)</p>	 <p>Kafka Topic Authorization</p>	 <p>Custom Application Authorization</p>	 <p>Permit.io</p>	 <p>HTTP API Authorization in PHP</p>	 <p>Fairwinds Insights Configuration Validation Software</p>
 <p>Kubescape Kubernetes security posture scanner</p>	 <p>Kubernetes Authorization</p>	 <p>Ceph Object Storage Authorization</p>	 <p>OPAL (Open Policy Administration Layer)</p>	 <p>SPIRE</p>	 <p>AWS CloudFormation Hook</p>

Checkers & linters

- [conftest](#)
- [kics.io](#)

Checkers & linters

conftest

```
package main

import data.kubernetes

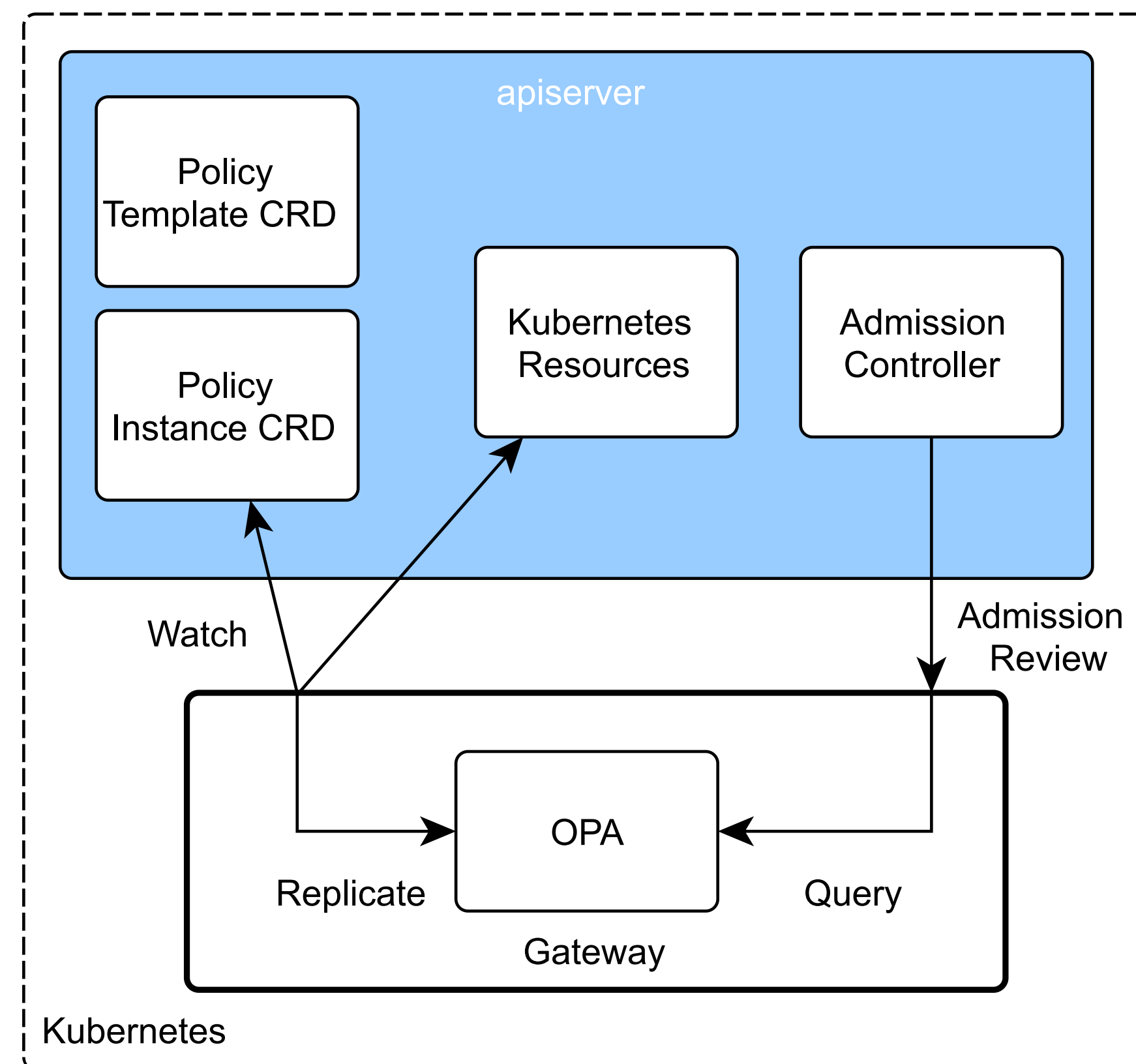
name = input.metadata.name

deny[msg] {
    kubernetes.is_deployment
    not input.spec.template.spec.securityContext.runAsNonRoot

    msg = sprintf("must not run as root in Deploy %s", [name])
}
```

Kubernetes

Admission hooks:



Kubernetes

```
package kubernetes.admission
operations = {"CREATE", "UPDATE"}

input_container[c] {
  c := input.request.object.spec.template.spec.containers[_]
}

deny[reason] {
  input.request.kind.kind == "Deployment"
  operations[input.request.operation]
  input_container[container]
  not container.resources.limits.cpu
  reason := sprintf("no cpu limit for container %v", [container])
}
```

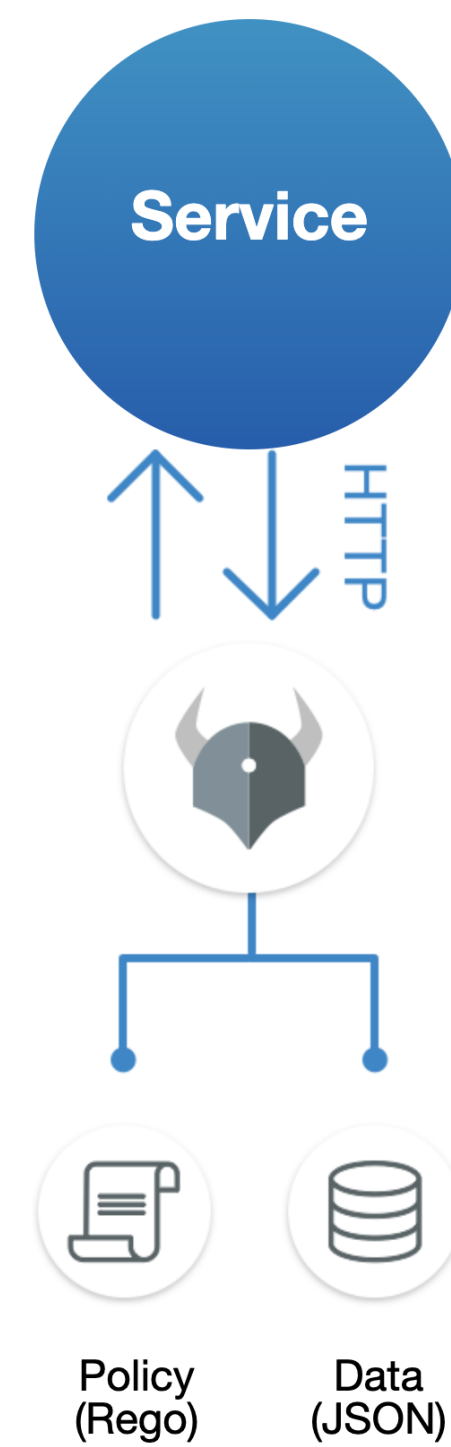
Kubernetes

OPA integrations:

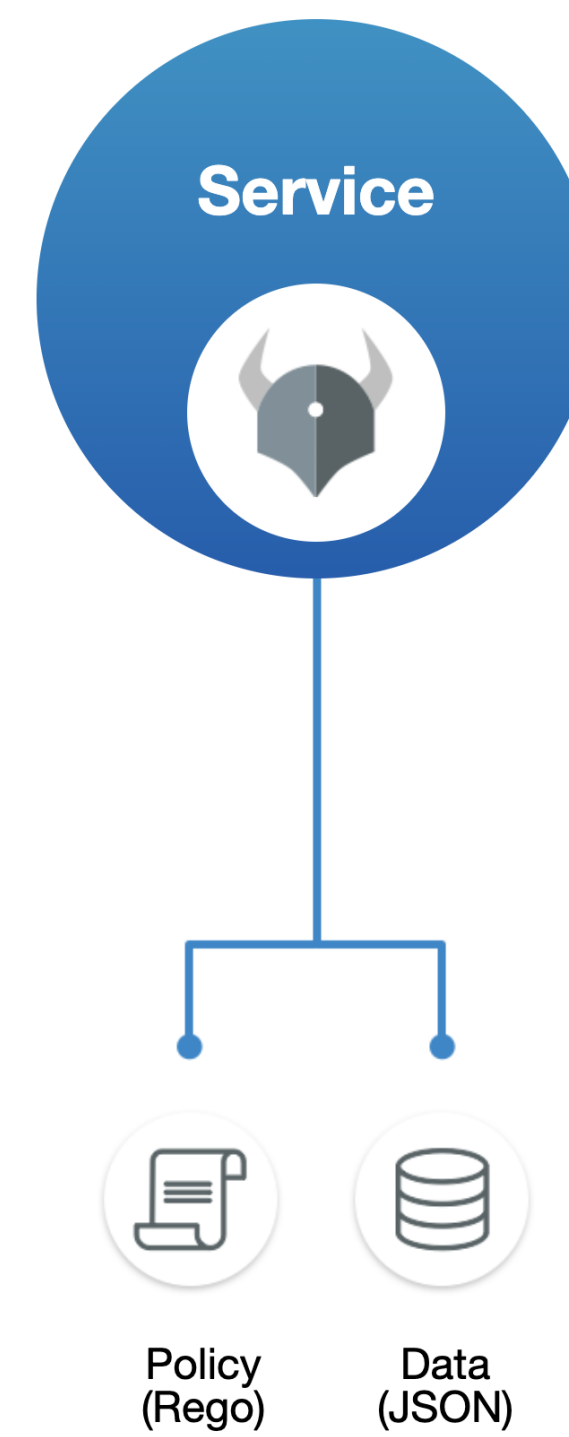
- Istio
- Envoy

Your app/tool

Daemon



Library



Golang & OPA

- Golang lib - embed OPA
- Golang SDK - high level API

See [docs](#)

Golang & OPA

Let's dive into the code.

Golang & OPA

```
var (  
    unsafeBuiltins = map[string]struct{}{  
        "http.send":          {},  
        "opa.runtime":         {},  
        "rego.parse_module":    {},  
        "time.now_ns":          {},  
        "trace":                {},  
    }  
)
```

```
rego.UnsafeBuiltins(unsafeBuiltins)
```

Golang & OPA

You can also add your custom functions for your customer to use, see [docs](#).

Lessons learnt

Open Policy Agent:

- easily integrates with Golang
- implement policy-as-a-code in a safe way
- not only for policies

Lessons learnt

Rego requires some time to master:

1. Policy workbench with samples
2. Examples and docs
3. Policy testing

Lessons learnt

Styra, for example, provides an interactive builder:

- Policy Builder

Alternatives

- [cuelang](#)
- [kyverno](#)
- [Sentinel](#)



Slides and code:

github.com/wojciech12

Wojciech Barczynski
wojciechb@spacelift.io



Hiring

(Go) Backend and Frontend developers
with a passion for building tools
for other engineers.