INFRACLOUD

API Authorization with Open Policy Agent

Gaurav Gajkumar Chaware

27th Jun, 2020

https://twitter.com/GGCTwts

Speaker

Gaurav Gajkumar Chaware

Senior Technology Architect

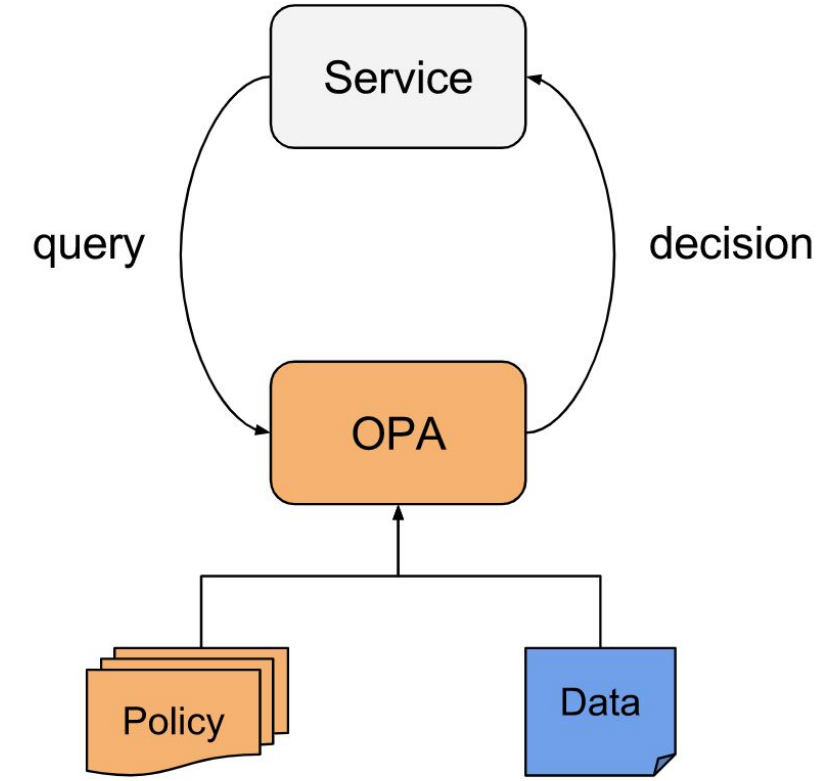
https://www.linkedin.com/in/gauravchaware

https://twitter.com/GGCTwts

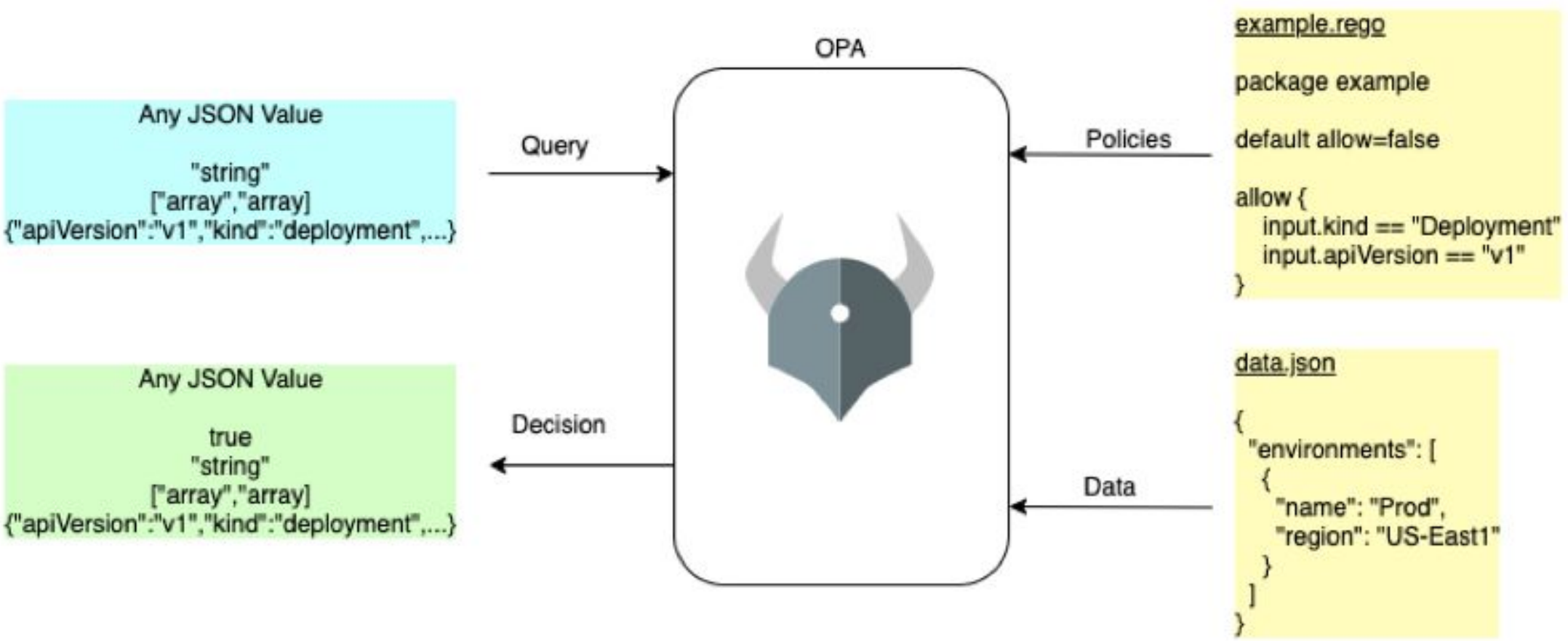
https://twitter.com/infracloudio

Open Policy Agent

- Open Source, General Purpose Policy Engine
- Run as
  - Server
  - Go Library
  - Interactive Shell or REPL
- Declarative Policy Language - Rego
- Policy As Code



Open Policy Agent



AuthN & AuthZ

- **AuthN** : Authentication : Who is accessing the Resource ? Identity.

- **AuthZ** : Authorization :
  - Is the user allowed to access this Resource?
  - What Operations can the user perform on this Resource?

AuthZ Policies for APIs

- **I**dentity can / cannot perform an **O**peration on a **R**esource
- In reality, AuthZ decision needs more contextual data.
  - Sometimes of the user -
    - What role does the user have?
    - Which Geographical region does user belong to?
  - Other times of the resource -
    - Which Geographical region does resource belong to?
    - Is it a restricted resource?

AuthZ Implementation in APIs

```
handleGetOffers() {
    ...
    if (authorize()) {
        //allow operation
        ...
    }
}


func authorize() {
    allow = false;
    if (user.Role == "admin") {
        allow = true;
    }

    if (user.Roles == "member") {
        if (resource.state == "new") {
            allow = true;
        }
    }

    return allow;
}
```

```
handleGetCustomers() {
    ...
    if (authorize()) {
        //allow operation
        ...
    }
}


func authorizeCustomerAccess() {
    allow = false;
    if (user.Role == "admin") {
        allow = true;
    }

    if (user.Roles == "member") {
        if (resource.state == "new") {
            allow = true;
        }
    }

    return allow;
}
```

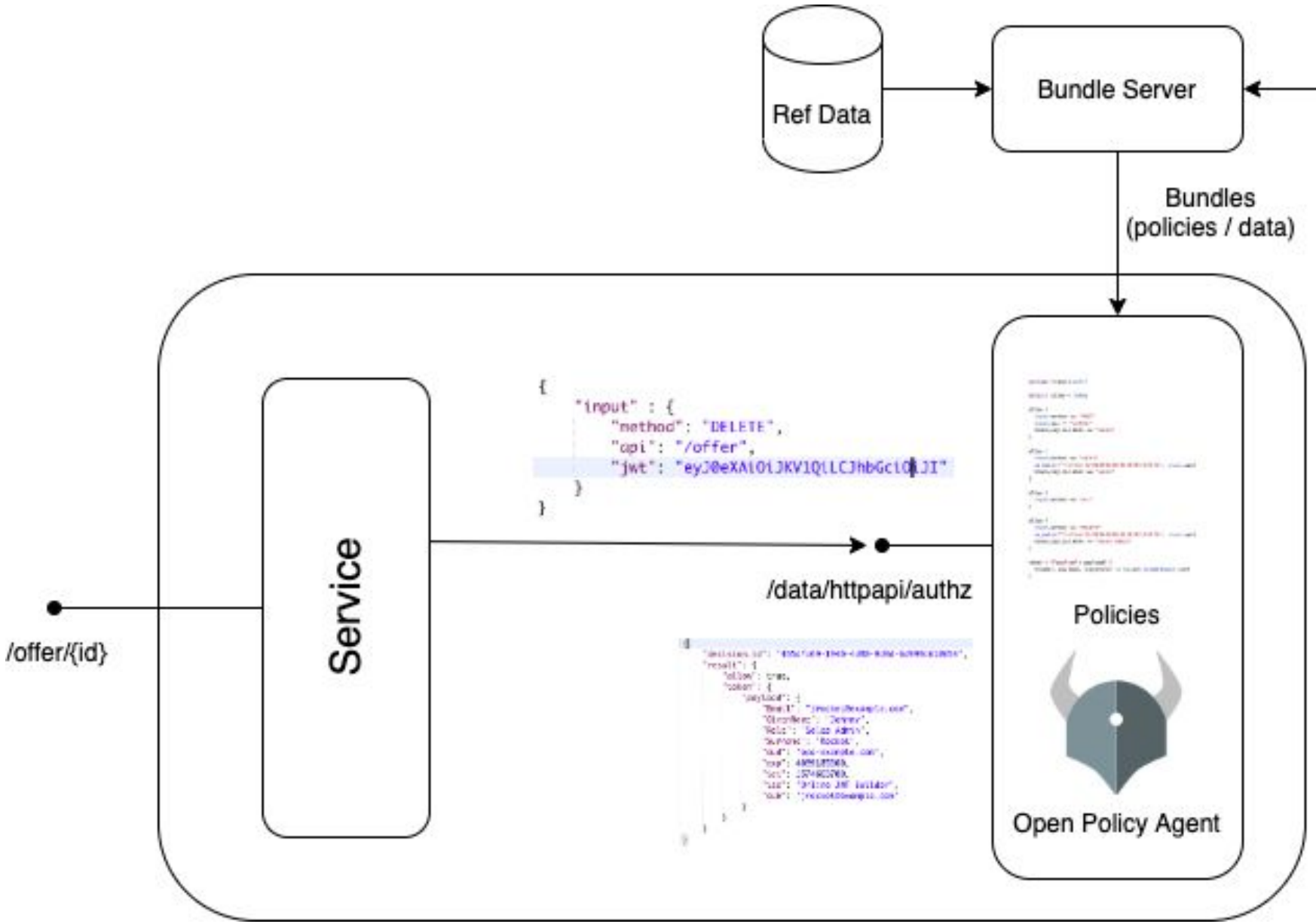
```
handleGetQuotes() {
    ...
    if (authorize()) {
        //allow operation
        ...
    }
}


func authorizeQuotesAccess() {
    allow = false;
    if (user.Role == "admin") {
        allow = true;
    }

    if (user.Roles == "member") {
        if (resource.state == "new") {
            allow = true;
        }
    }

    return allow;
}
```

AuthZ solution for APIs

- Decouple policies from Application Code
- Shared / Universal policies
- Common language across domains
- Centralized policy management
- Centralized logging and auditing of policy enforcement

Demo

- A sample CPQ application where sales team creates quotations



Use OPA as a :

- Sidecar in your pod
- Standalone service
- Golang Library

OPA Configuration

- **Bundle Service :** manage policies & data centrally
- **Discovery :** manage OPA configurations centrally
- **Decision Telemetry :** Report decisions to remote HTTP service
  - Mask sensitive data from decision logs
- **Agent Telemetry :** Report OPA agent status to remote HTTP Service
- **Monitoring :** HTTP Endpoint for prometheus metrics

OPA Ecosystem

- https://www.openpolicyagent.org/docs/latest/ecosystem/

- OPA unifies policy enforcement across the stack
- Common policy language across diverse set of technologies

Open House & Thank you!

INFRACLOUD

ધન્યવાદ!

Dank je!

Kiitos!

આભાર!

धन्यवाद!

Grazie!

Je vous remercie!

ありがとうございました！

ਤੁਹਾਡਾ ਧੰਨਵਾਦ!

நன்றி!

ధన్యవాదాలు!

ನಂತಿ!

THANK YOU!

Follow us https://twitter.com/infracloudio