OPEN POLICY AGENT

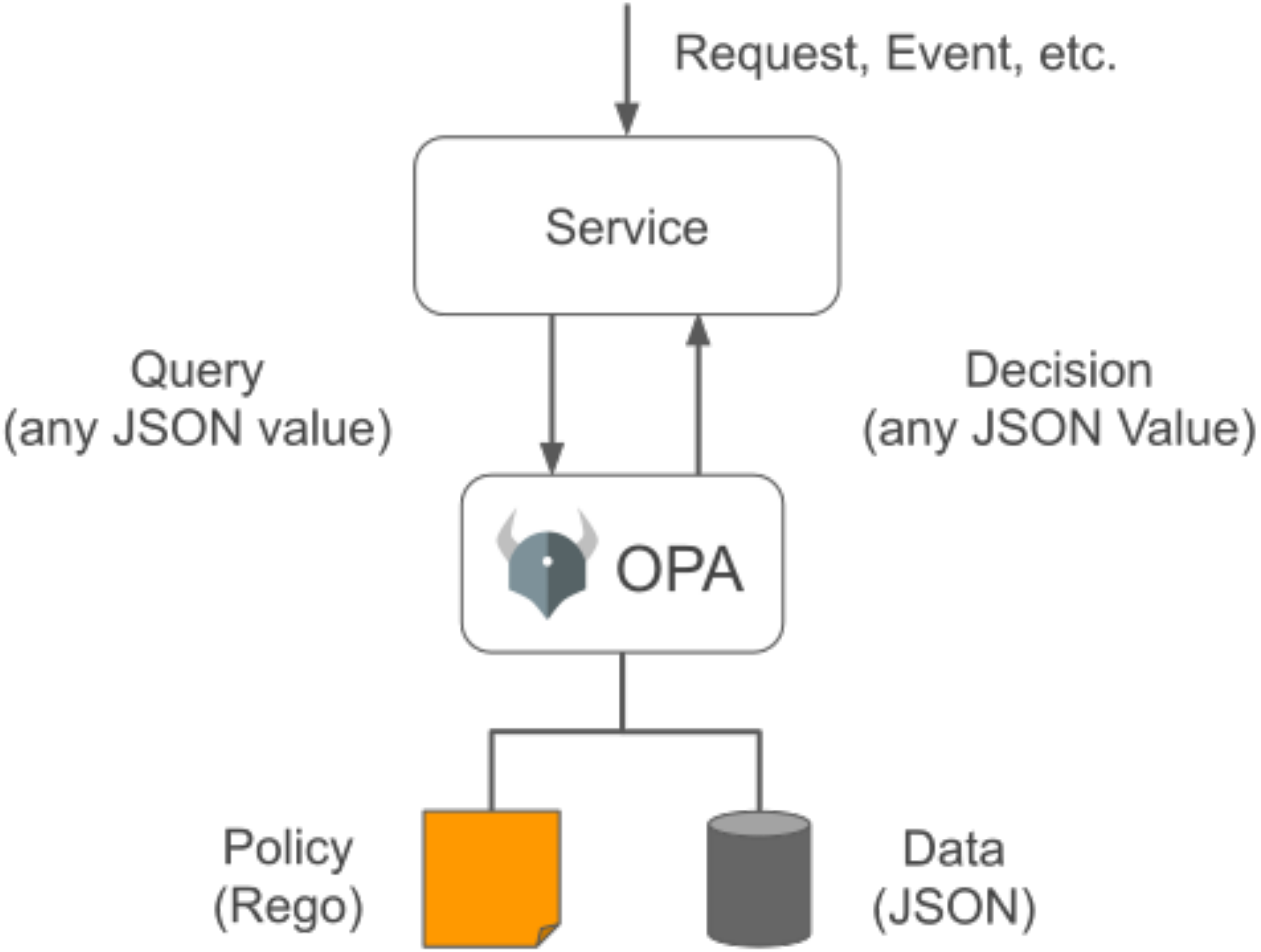
Introduction and Short Demo

KESHAV PRASAD

INTRODUCTION

- Open policy agent is a general purpose policy engine
- Policy as code
- Declarative model
- Offload decision making from source code to Opa
- Simple code language - Rego
- Can be used in Micro Services, Kubernetes, API gateways etc
- Written in Golang and is super fast and super efficient
- Open Source CNCF graduated project
  - https://www.cncf.io/projects/
  - https://www.openpolicyagent.org/

REQUEST FLOW



Request, Event, etc.

Service

Query
(any JSON value)

Decision
(any JSON Value)

OPA

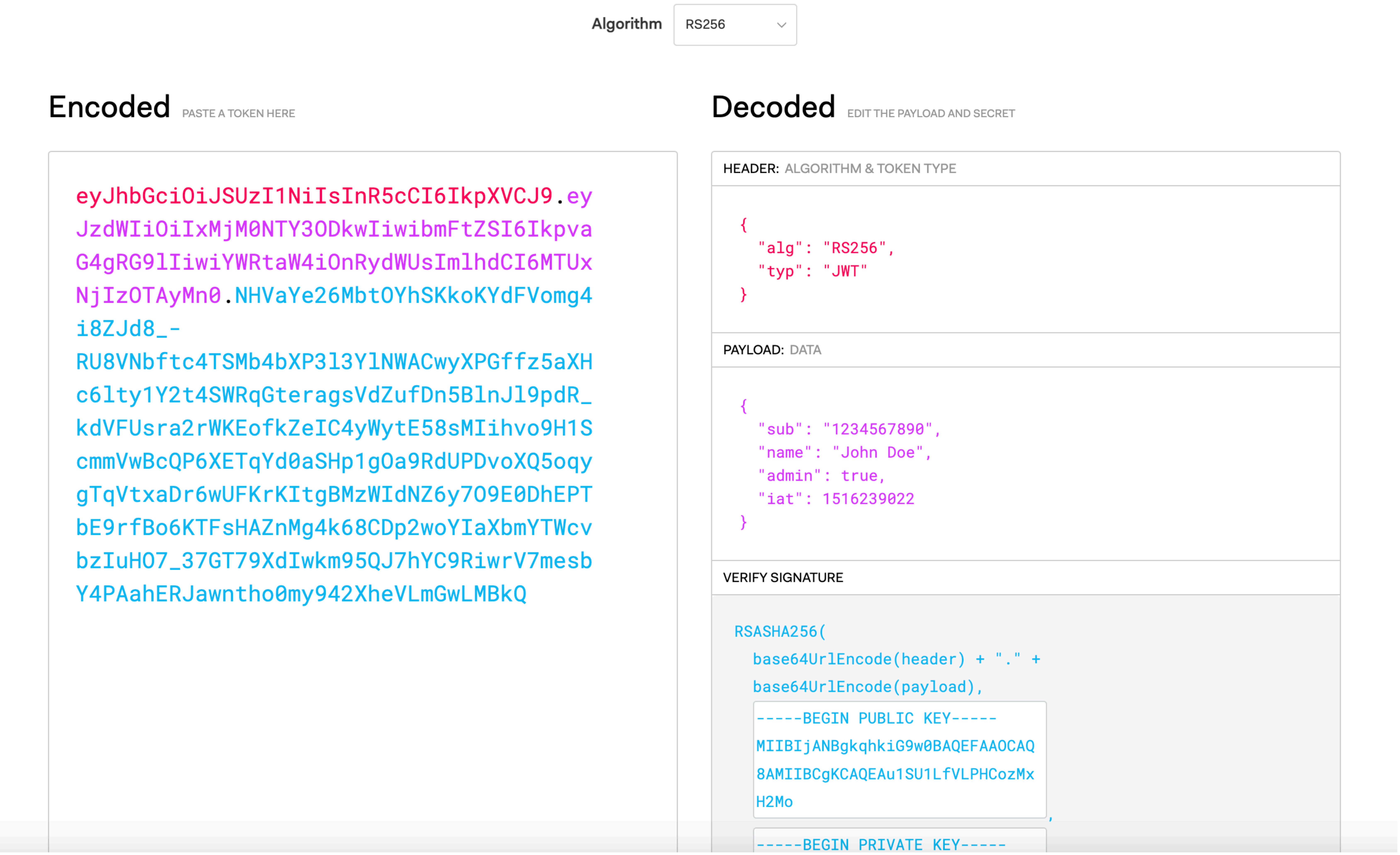
Policy
(Rego)

Data
(JSON)

JSON WEB TOKENS

- JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties
- JWTs is the most common way a client authenticates with a server
- A JWT usually contains information about the user
- A JWT token usually has an expiry
- JWT can be created with many algorithms such as RS256, RS512, HS256 etc
- A JWT consists of three parts
  - Header
  - Payload
  - Signature
- JWTs can be verified offline

SAMPLE JWT TOKEN - RS256 (RSA WITH SHA-256)



Algorithm: RS256

Encoded
PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiYWRtaW4iOnRydWUsImlhdCI6MTUx
NjIzOTAyMn0.NHVaYe26MbtOYhSKkoKYdFVomg4
i8ZJd8_-
RU8VNbftc4TSMb4bXP3l3YlNWACwyXPGffz5aXH
c6lty1Y2t4SWRqGteragsVdZufDn5BlnJl9pdR_
kdVFUsra2rWKEofkZeIC4yWytE58sMIihvo9H1S
cmmVwBcQP6XETqYd0aSHp1gOa9RdUPDvoXQ5oqy
gTqVtxaDr6wUFKrKItgBMzWIdNZ6y7O9E0DhEPT
bE9rfBo6KTFsHAZnMg4k68CDp2woYIaXbmYTWcv
bzIuHO7_37GT79XdIwkm95QJ7hYC9RiwrV7mesb
Y4PAahERJawntho0my942XheVLmGwLMBkQ
```

Decoded
EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

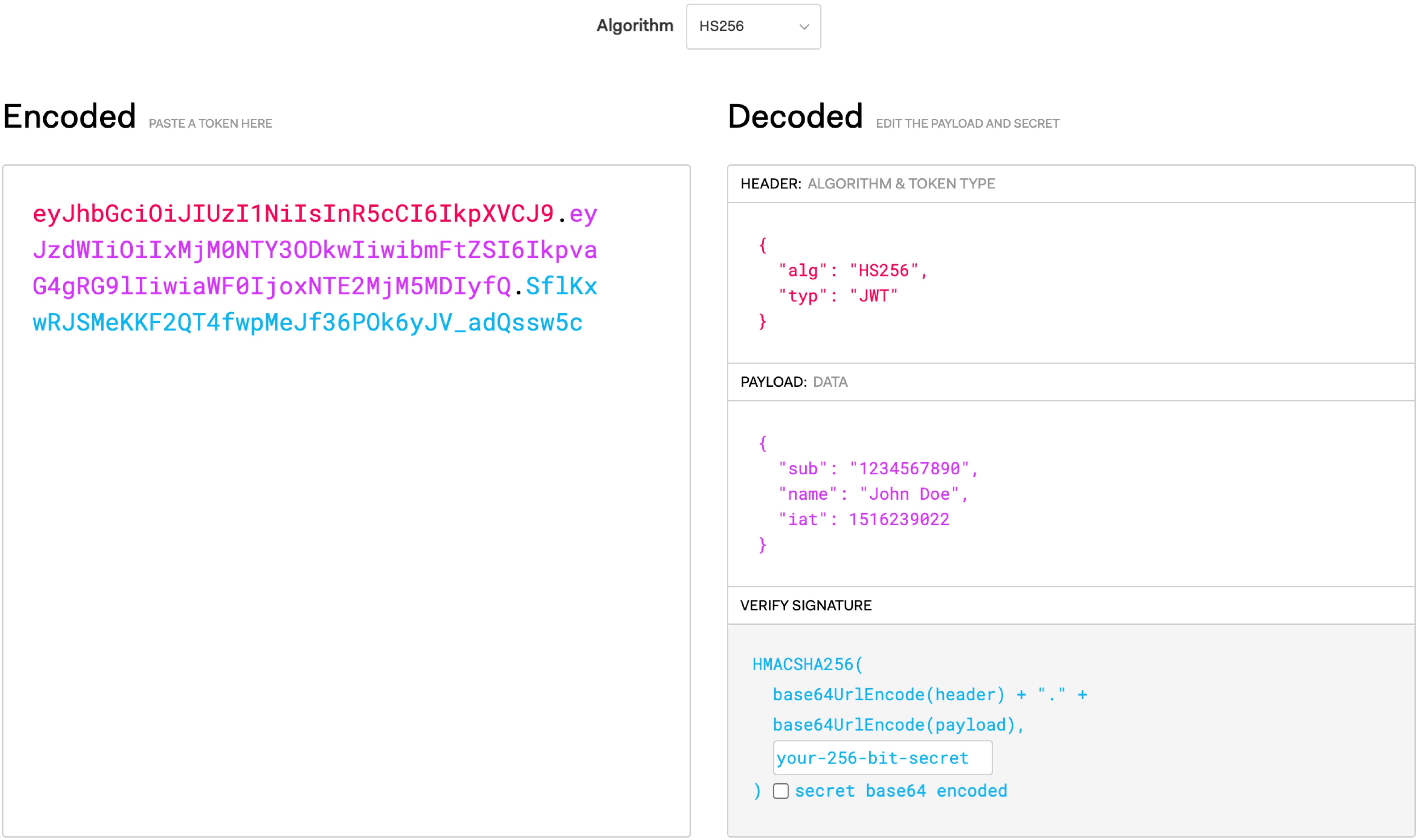
PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true,
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  -----BEGIN PUBLIC KEY-----
  MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ
  8AMIIBCgKCAQEAu1SU1LfVLPHCozMx
  H2Mo
  ,
  -----BEGIN PRIVATE KEY-----
```

SAMPLE JWT TOKEN - HS256 (HMAC WITH SHA-256)



Algorithm HS256

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) □ secret base64 encoded
```

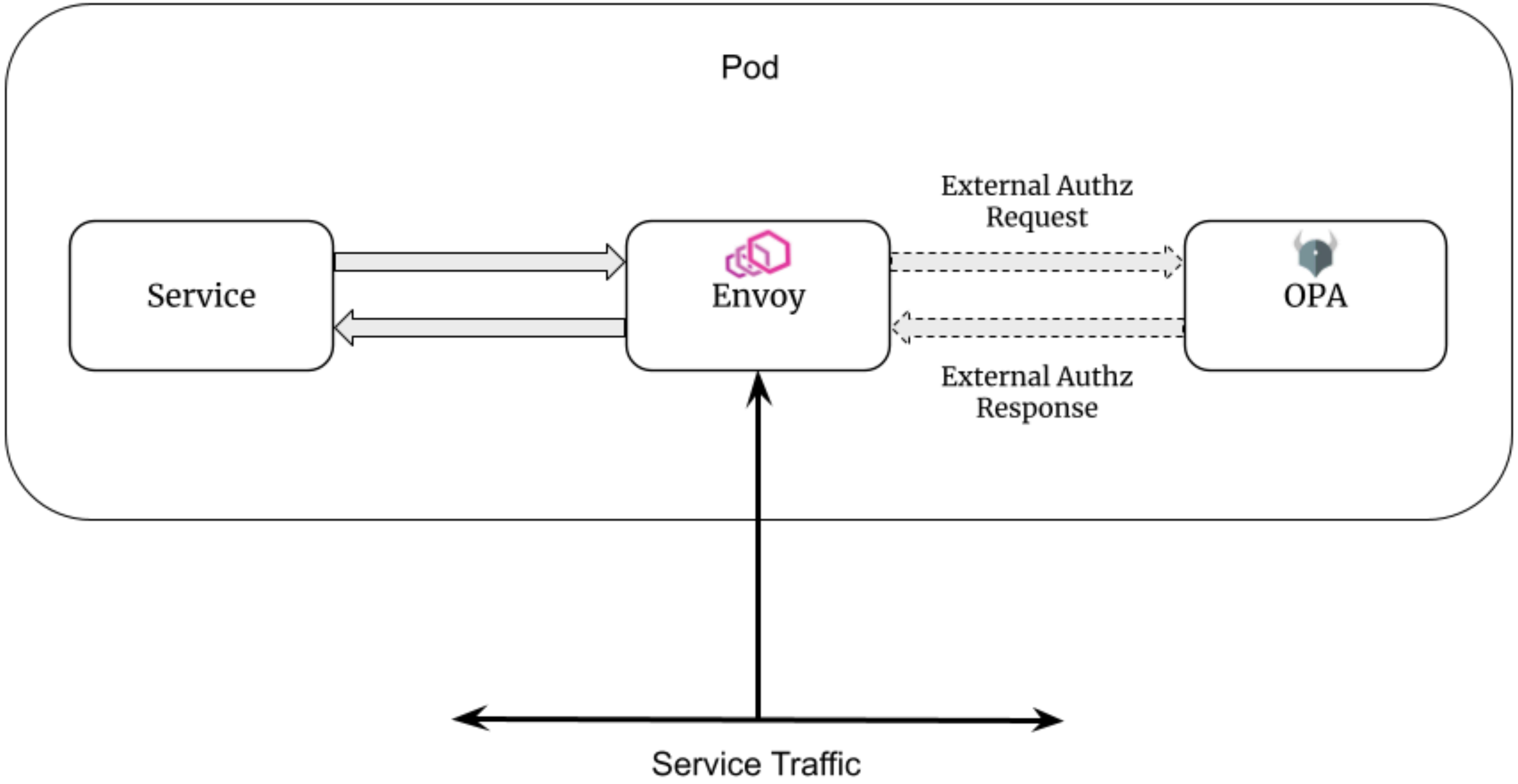
RUNNING OPA

- OPA is a golang binary that can be downloaded and run without any installation required
- Works on Linux, MacOS, Windows
- Check the download instructions here
  - https://www.openpolicyagent.org/docs/latest/#running-opa

OPA DOCUMENTS AND FUNCTIONS

- data
- package
- built in functions
- input
- Check out the below URL for more information, commands and hands on
  - https://github.com/keshavprasadms/opa-demo/blob/main/README.md

OPA ARCHITECTURE ON KUBERNETES


Pod
External Authz
Request
Service
Envoy
OPA
External Authz
Response
Service Traffic