Using SPIFFE & OPA to Authenticate & Authorize Workloads

Presented By Charlie Egan

Hello

Charlie Egan

Developer Advocate, Styra

- OPA Maintainer, contributing since 2019
- Interested in SPIFFE & all things authn/authz
- Joined Styra in December 2022, previously at Jetstack

Email: charlie@styra.com
Mastodon: charlieegan3@hachyderm.io
Twitter: @charlieegan3
LinkedIn: charlieegan3
Website: charlieegan3.com

What is a ‘Workload’?

What is a ‘Workload Identity’?

What is a ‘Good Workload Identity’?

What is a ‘Good Workload Identity’?

A ‘Good Workload Identity’...

…is short lived and automatically rotated

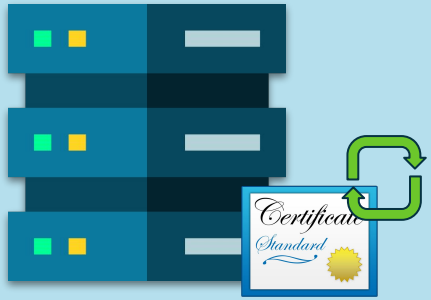


image credit: flaticon.com

A ‘Good Workload Identity’...

…can’t be captured in transit and replayed by a bad actor.

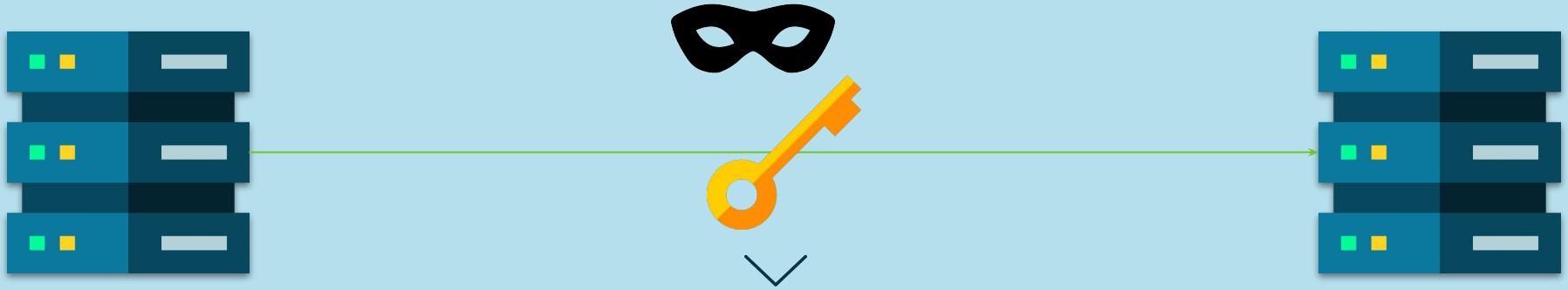


image credit: flaticon.com

A ‘Good Workload Identity’...

Trust for a ‘Good Workload Identity’ can be carefully bounded

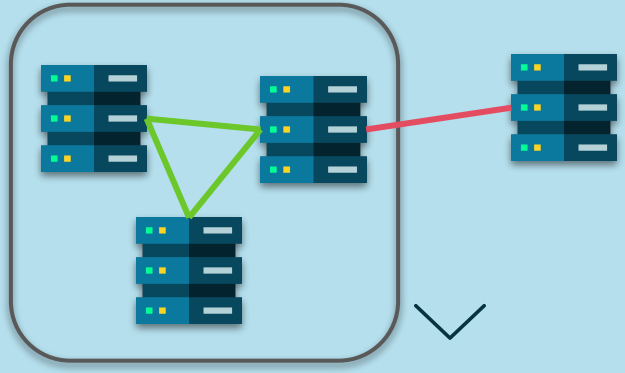


image credit: flaticon.com

A ‘Good Workload Identity’...

…is known by the invoked service

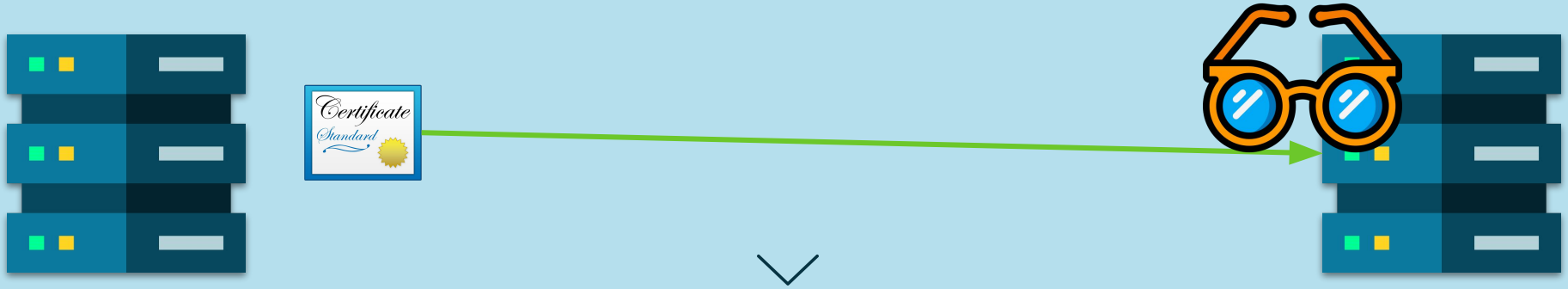


image credit: flaticon.com

Workload Identity Showdown

	Shared Secrets	Publicly trusted certs & mTLS	Service Meshes	?
Short Lived	❌	❌	✅	✅
Not Replayable	❌	✅	✅	✅
Scope is Bounded	❌	❌	✅	✅
Known by service	✅	✅	❌ / maybe	✅

Workload Identity Showdown

	Shared Secrets	Publicly trusted certs & mTLS	Service Meshes	SPIFFE mTLS
Short Lived	❌	❌	✅	✅
Not Replayable	❌	✅	✅	✅
Scope is Bounded	❌	❌	✅	✅
Known by server	❌	✅	❌ / maybe	✅

Meet SPIFFE

	SPIFFE mTLS
Short Lived	✅
Not Replayable	✅
Scope is Bounded	✅
Known by server	✅

spiffe

Secure Production Identity Framework for Everyone

Meet SPIFFE

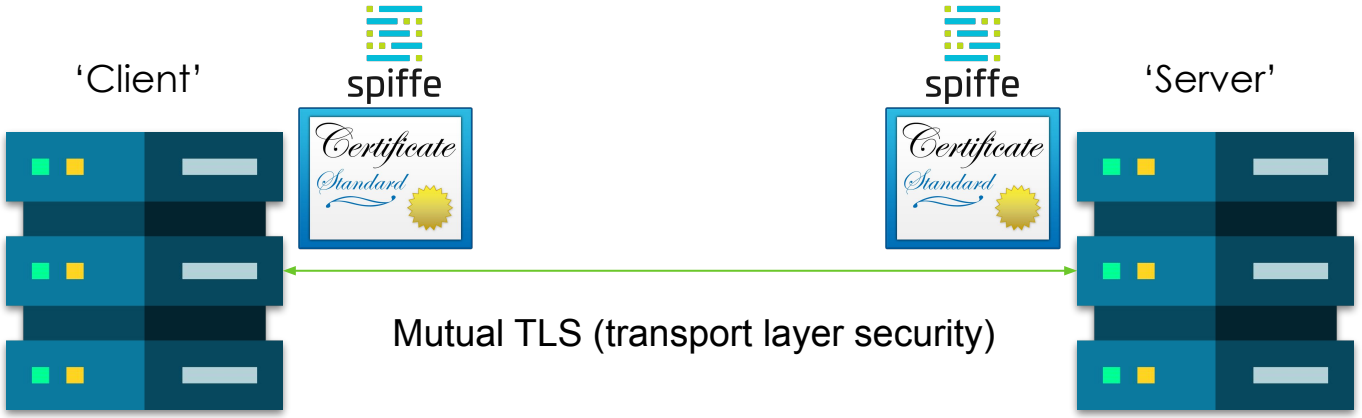
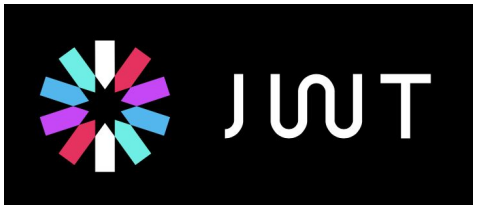


image credit: flaticon.com

Meet SPIFFE



Meet SPIFFE

```
Serial number:     4520694392272087521 (0x3ebcbb991d7a69e1)
Algorithm ID:      SHA256withECDSA
...
Subject
  O             = Example
  serialNumber = 4520694392272087521
Extensions
  keyUsage CRITICAL:
    digitalSignature
  extKeyUsage :
    clientAuth, serverAuth
  subjectAltName :
    uri:spiffe://example.com/foo/bar/baz
```


spiffe://example.com/foo/bar

SPIFFE IDs, now what?

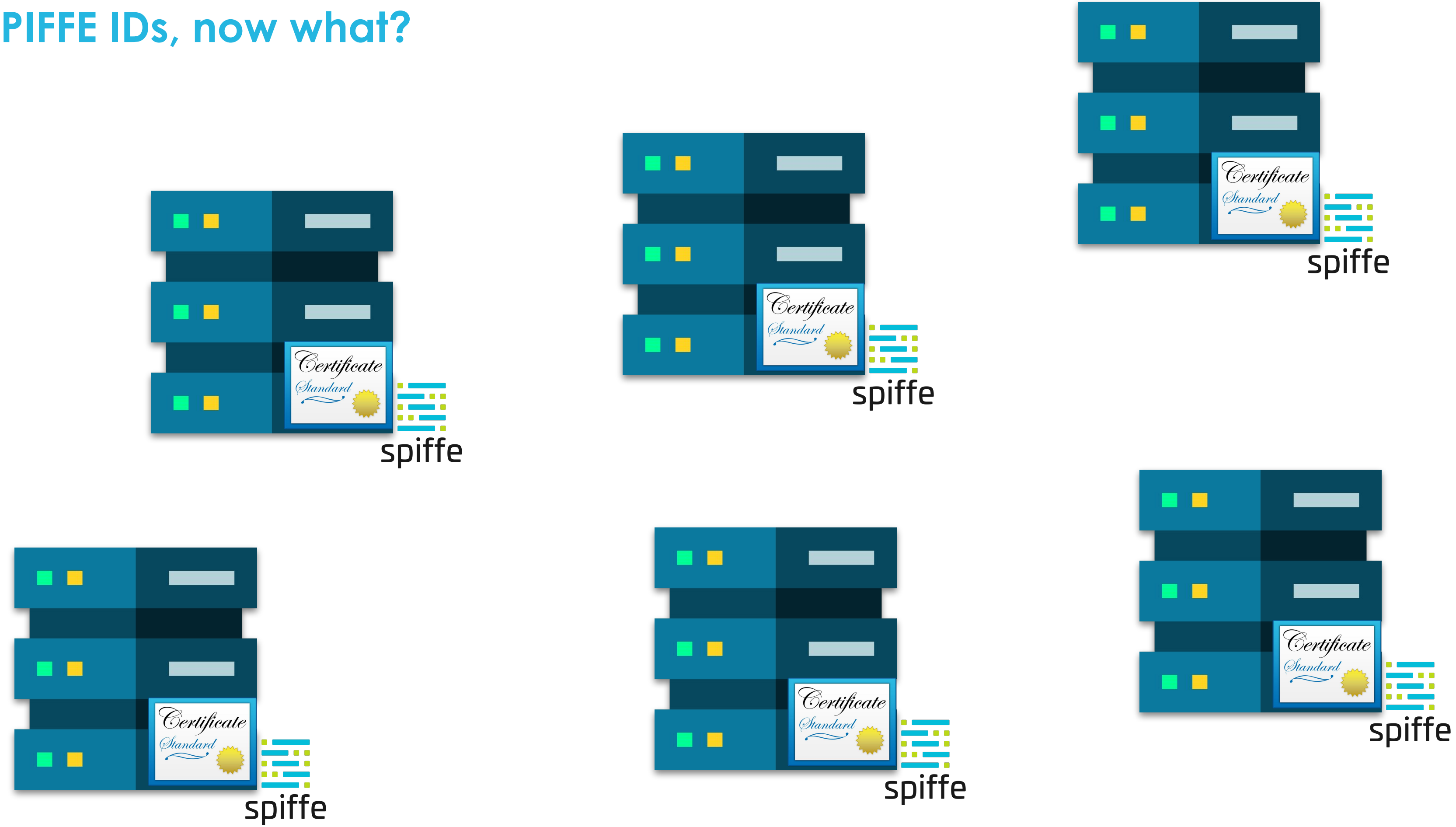


image credit: flaticon.com

 |

SPIFFE IDs, now what?

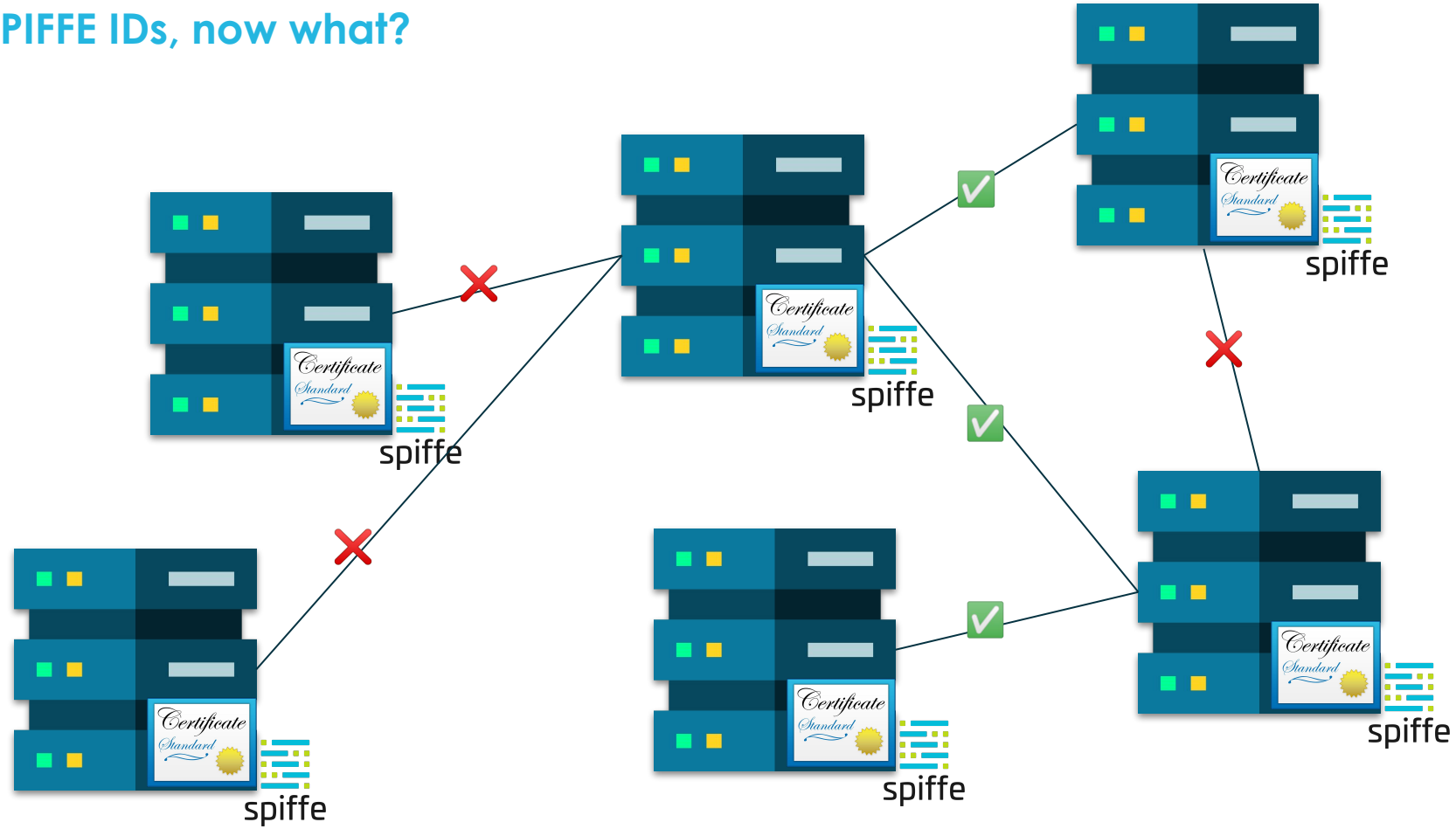


image credit: flaticon.com

SPIFFE IDs, now what?

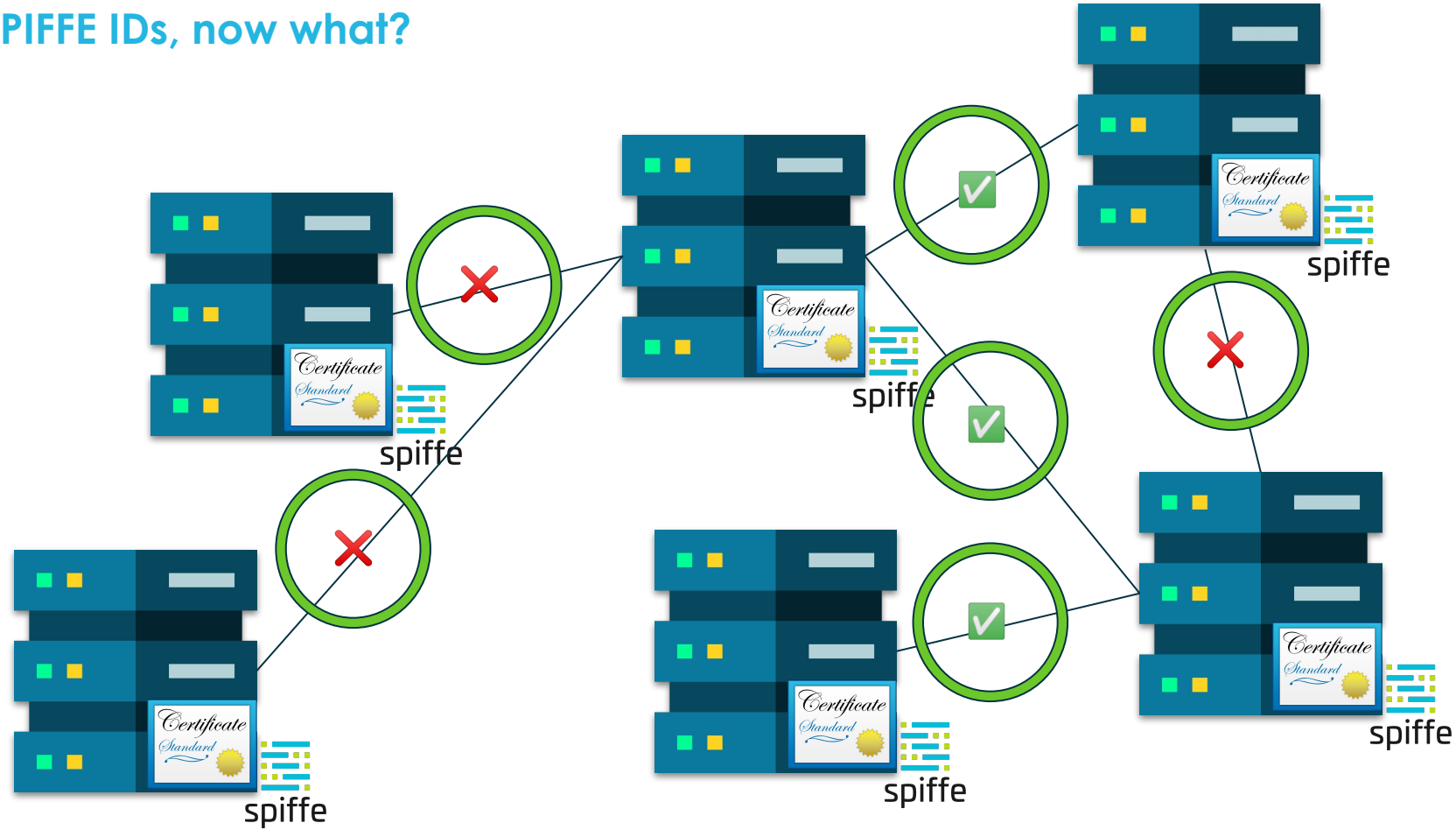


image credit: flaticon.com

SPIFFE IDs, now what?

Meet OPA

Meet Rego

example.rego

```
1 package authz
2
3 import future.keywords.if
4
5 default allow := false
6
7 allow if input.user.role == "admin"
```

Input:

```
{
    "user": {
        "email": "alice@example.com",
        "role": "admin"
    }
}
```


Meet OPA

Example Request:

```
POST /v0/data/authz/allow HTTP/1.1
Content-Type: application/json
{
    "user": {
        "email": "alice@example.com",
        "role": "admin"
    }
}
```

```
1 package authz
2
3 import future.keywords.if
4
5 default allow := false
6
7 allow if input.user.role == "admin"
```

Example Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
true
```


SPIFFE, OPA now what?

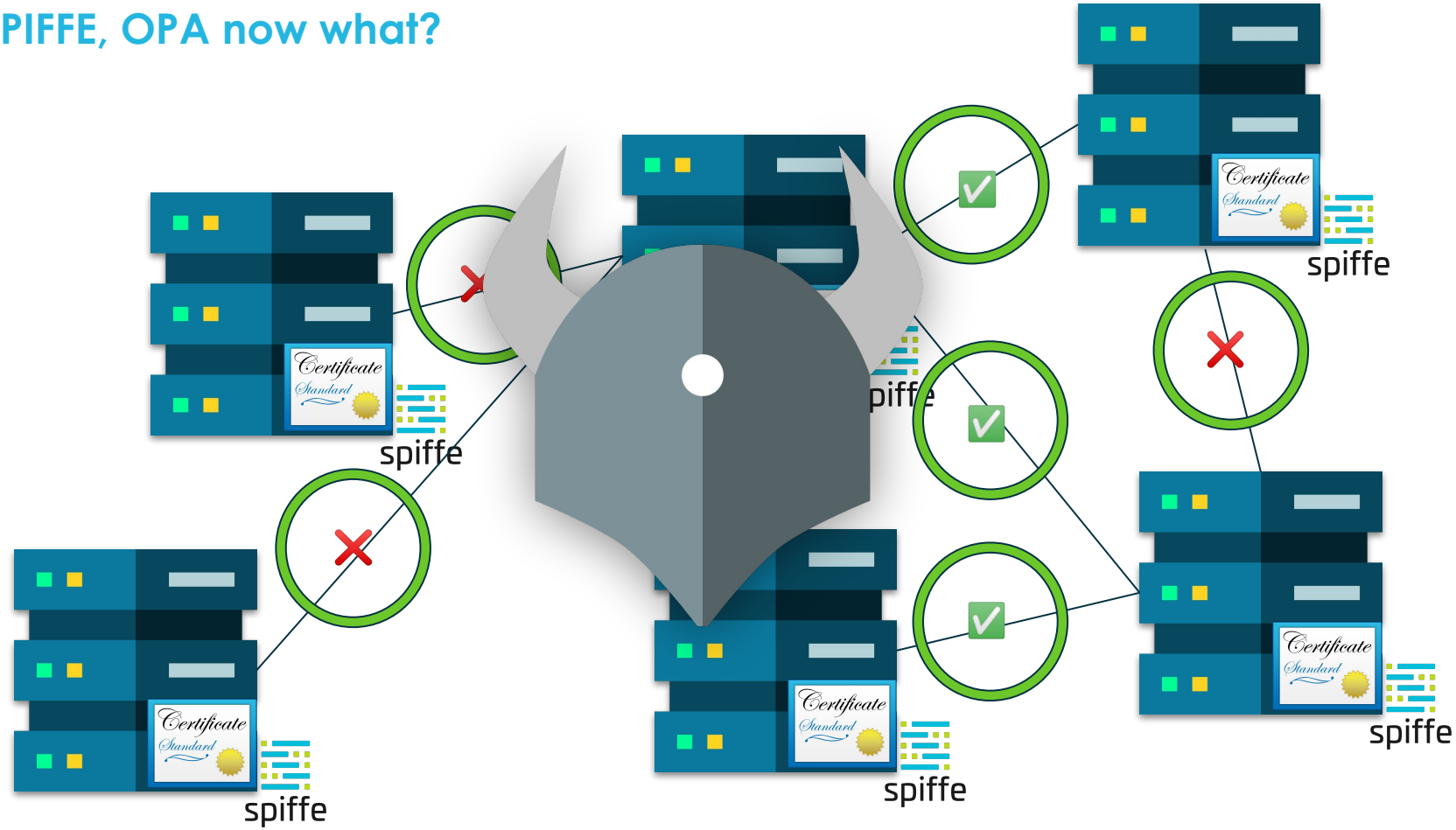
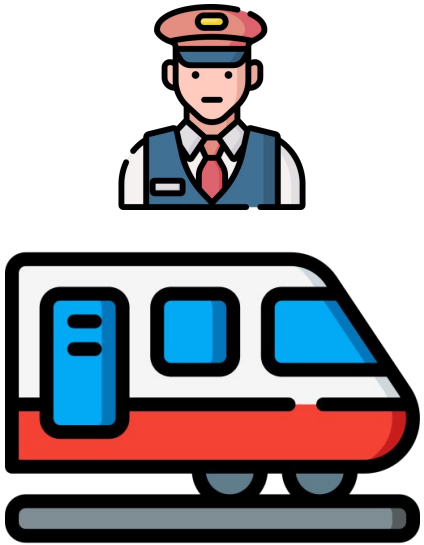


image credit: flaticon.com

Demo Environment Scenario

Demo Environment Scenario

Train Driver in Train



Local Station

HQ

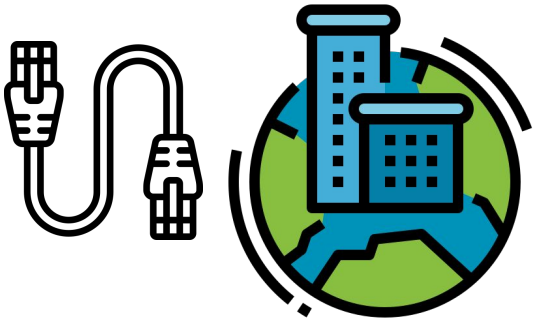


image credit: flaticon.com

Demo Environment Scenario

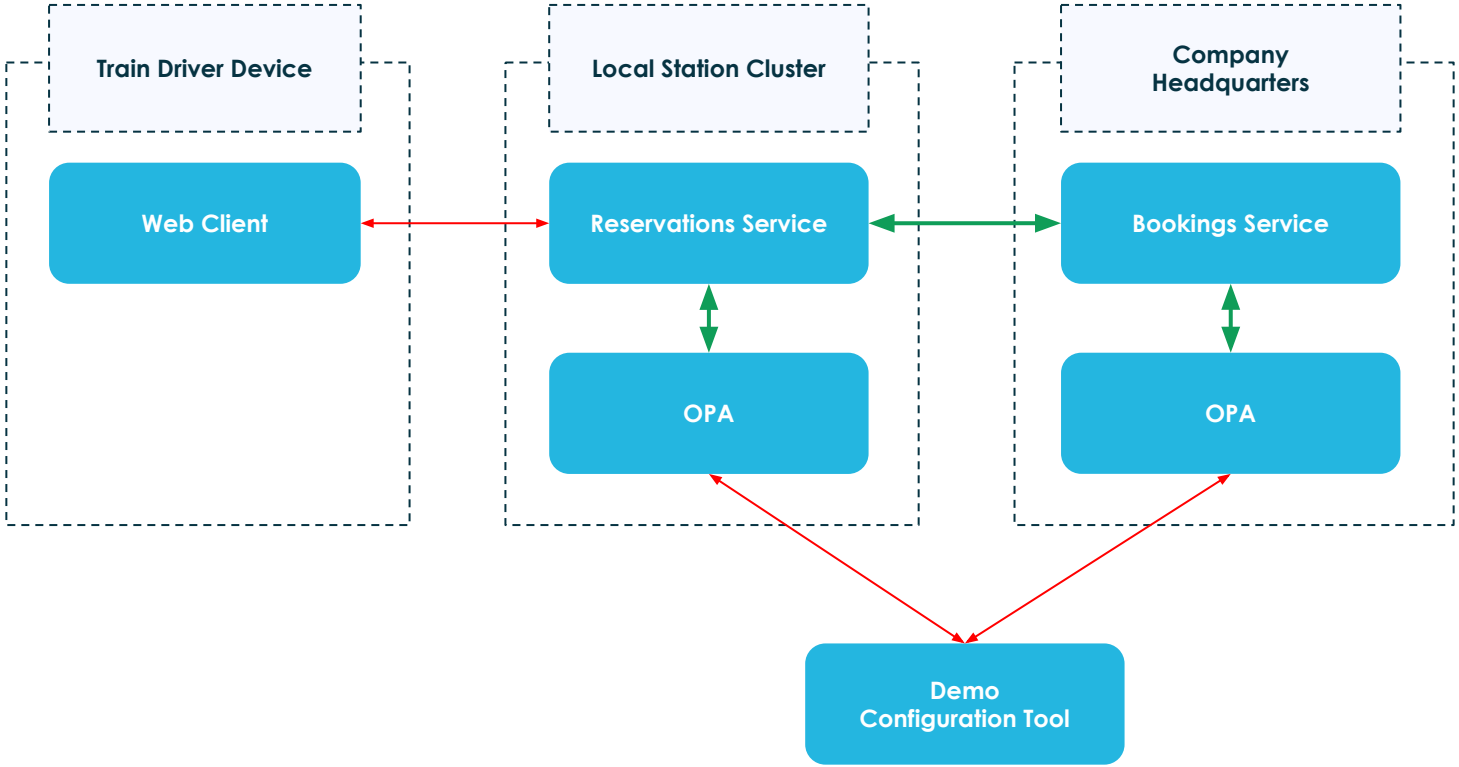
Train Driver in Train

Local Station

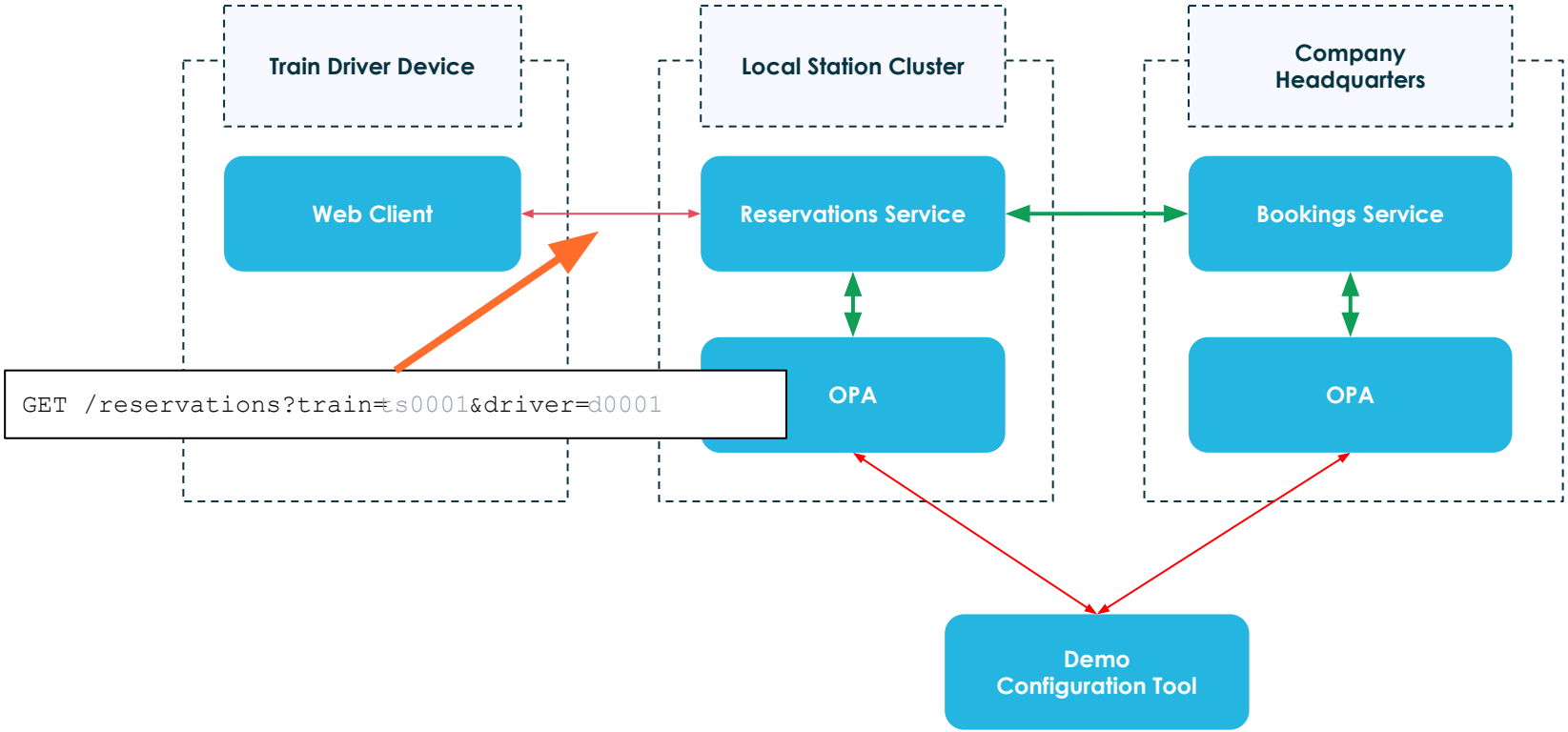
HQ

image credit: flaticon.com

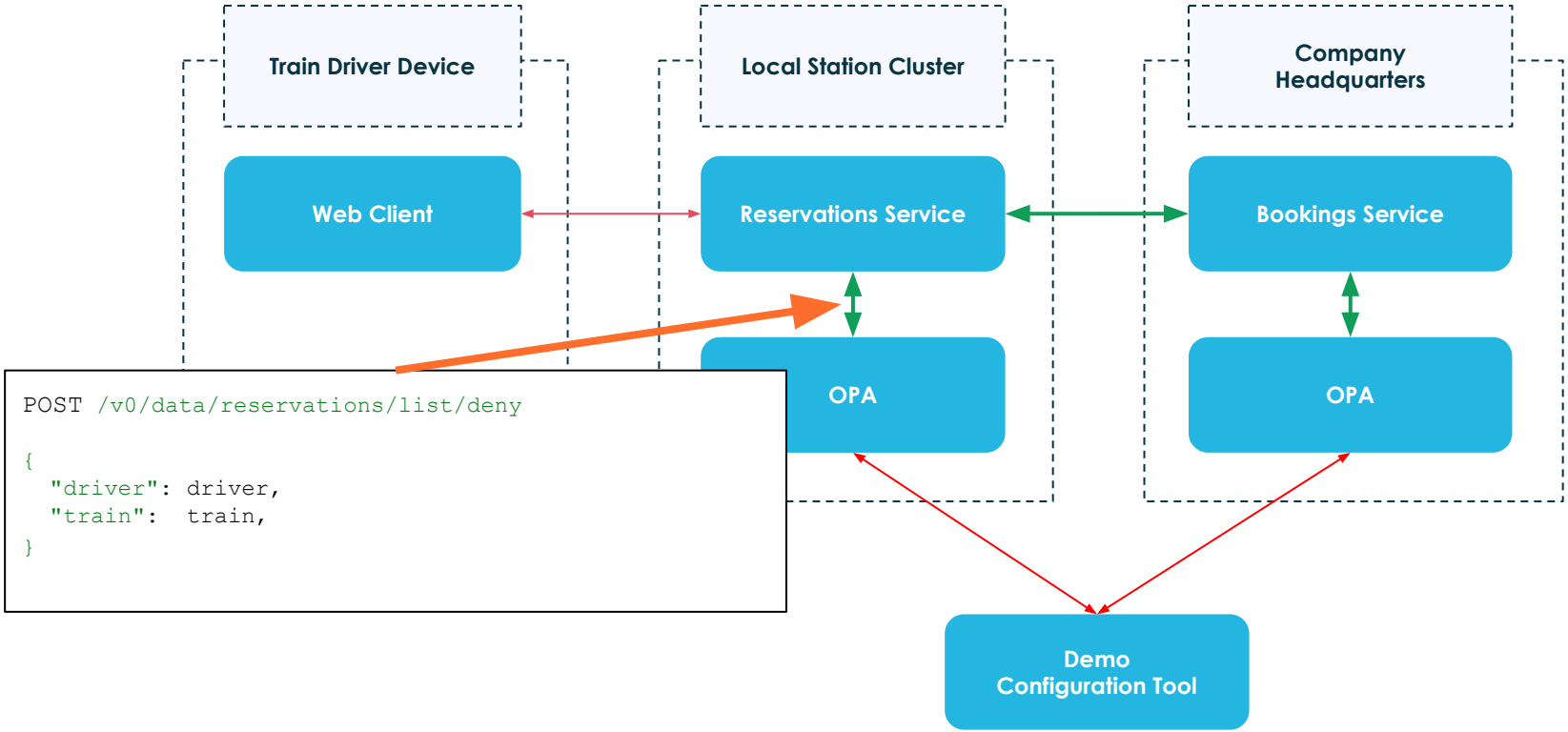
Demo Environment Architecture



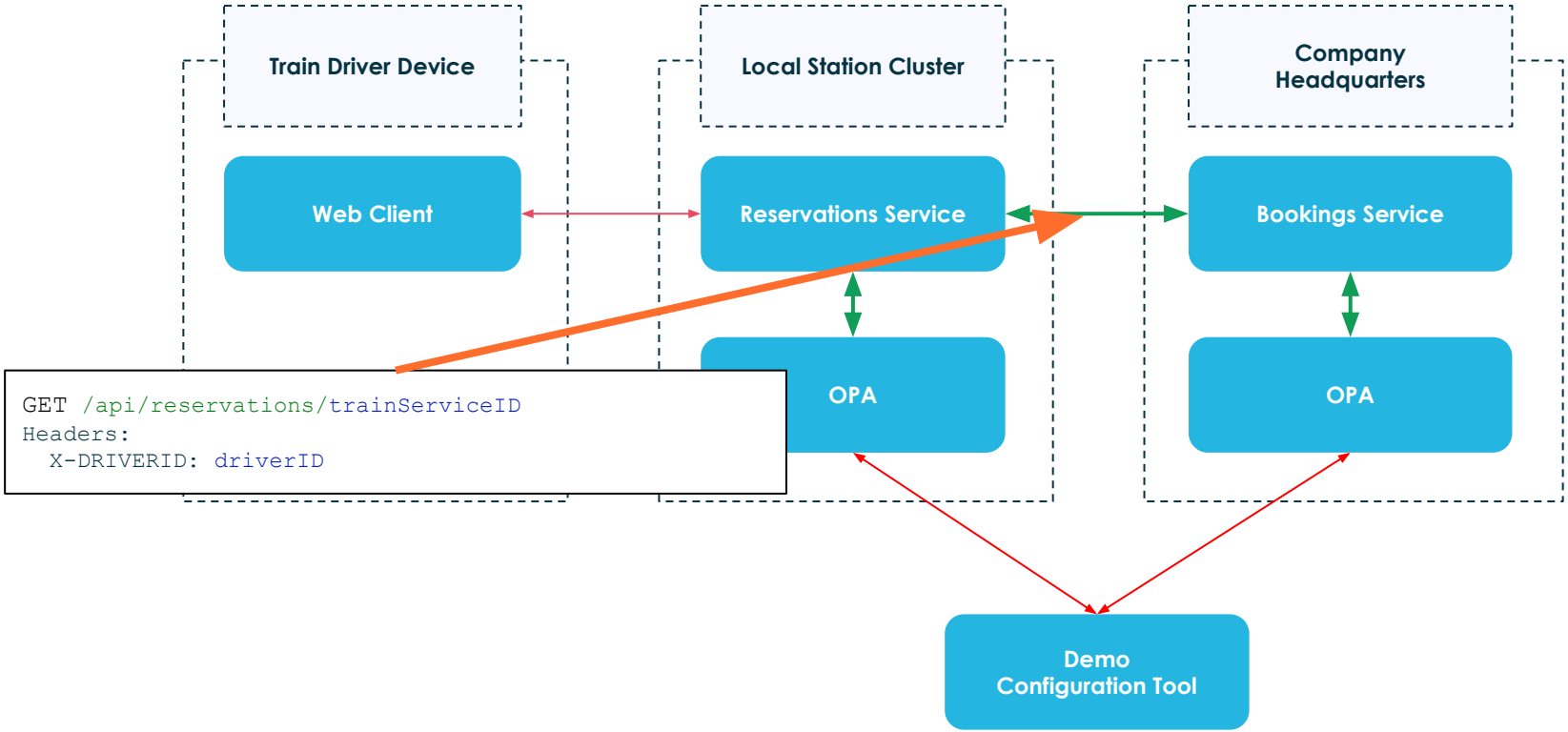
Demo Environment Architecture



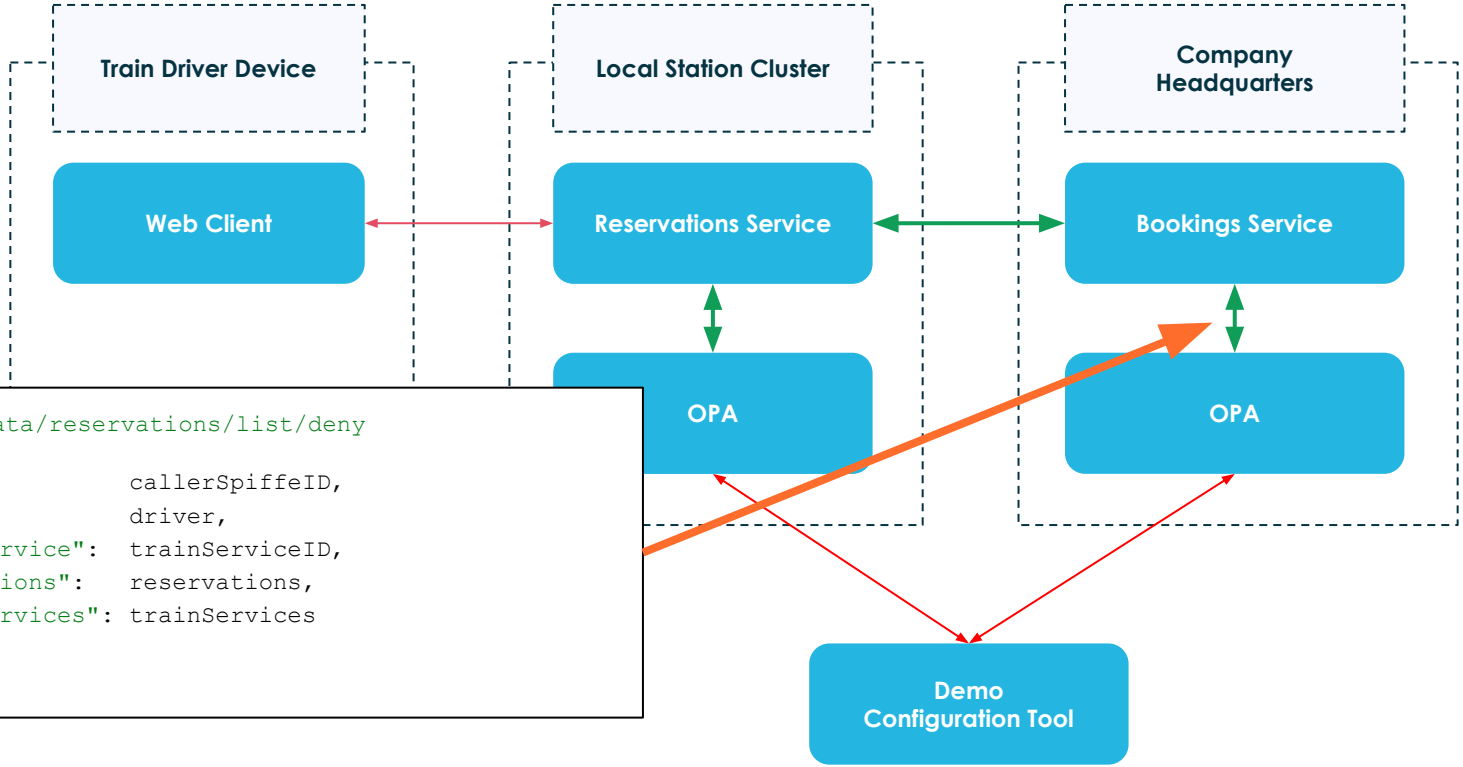
Demo Environment Architecture



Demo Environment Architecture

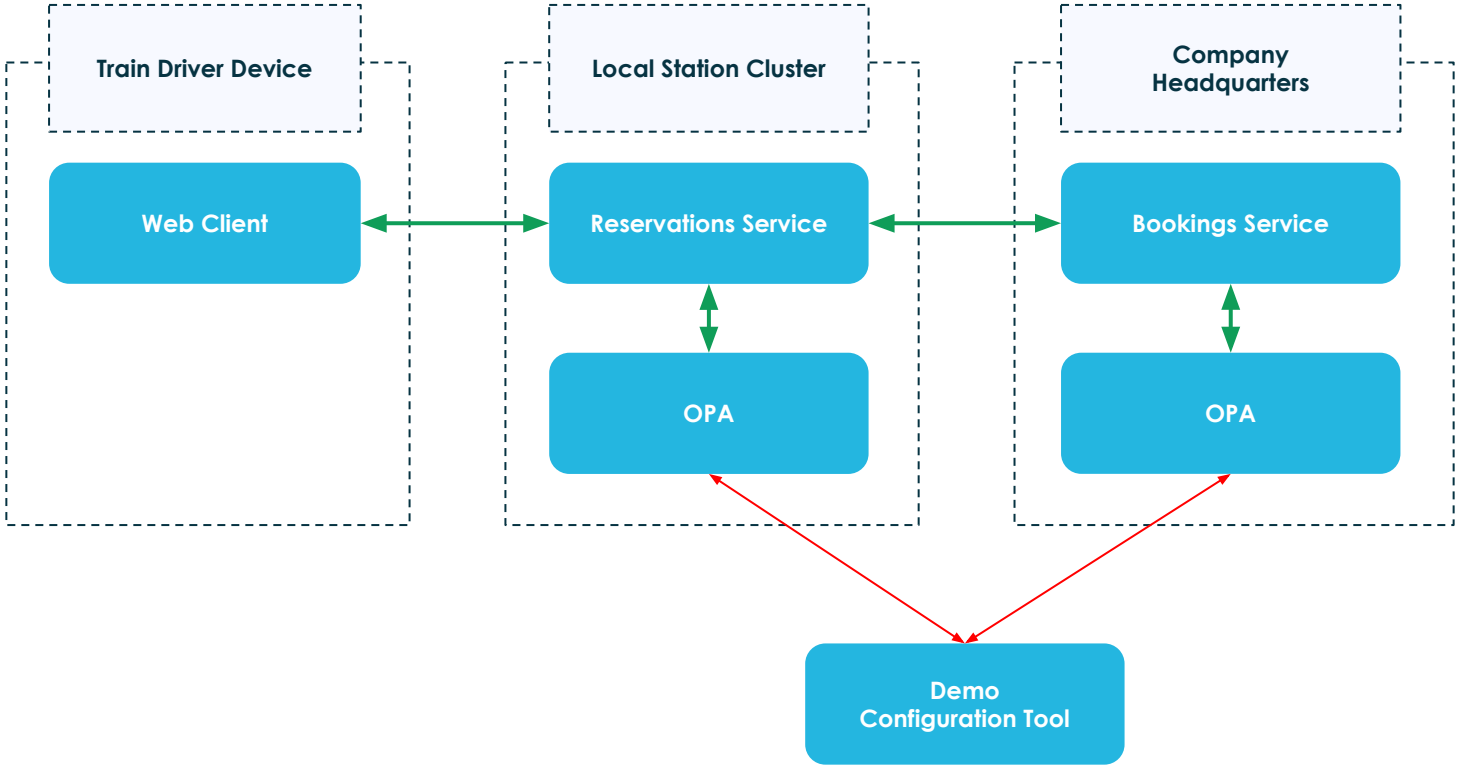


Demo Environment Architecture




```
POST /v0/data/reservations/list/deny
{
  "caller":         callerSpiffeID,
  "driver":         driver,
  "train_service":  trainServiceID,
  "reservations":   reservations,
  "train_services": trainServices
}
```


Demo Environment Architecture



Demo Summary

In this demo we have seen:

- SPIFFE IDs used in OPA authorization policies
- OPA used to enforce access
- OPA used to control behavior

Smoke and mirrors…

- This demo was meant to be running in Kubernetes

Work done in OPA for this presentation

- Support authorizing callers with SPIFFE IDs to OPA

What's left to do in OPA

- Support loading OPA bundles over SPIFFE mTLS
- Perhaps Rego SPIFFE ID functions like `id_in_trust_domain`

Links

- Code: https://github.com/charlieegan3/talk-opa-spiffe
- Slides:
- Styra Products
  - https://www.styra.com/styra-das/
  - https://www.styra.com/styra-load/

Thank You