Hello

Charlie Egan

Developer Advocate, Styra

- OPA Maintainer, contributing since 2019
- Interested in SPIFFE & all things authn/authz
- Joined Styra in December, 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'?

https://www.jetstack.io › blog › workload-identity-wit... ⋮

Modern workload identity with SPIFFE & Trust Domains

Learn how to configure **SPIFFE** workload identities using cert-manager. ... We created cert-manger at **Jetstack** to make issuing certificates easier in ...
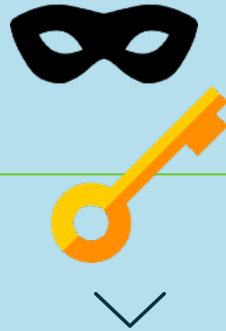
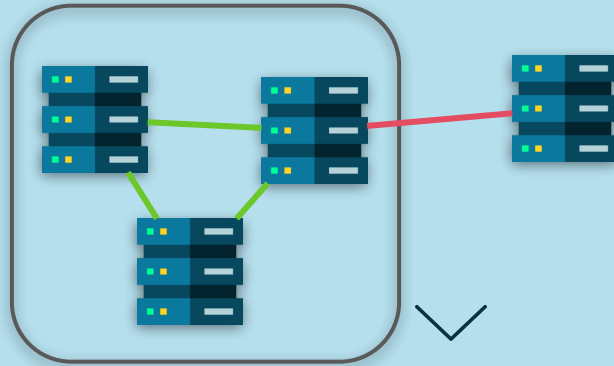# A '**Good**' Workload Identity'...

...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 service	✅	✅	❌ / maybe	✅

Meet SPIFFE

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

Secure Production Identity Framework for Everyone

Meet SPIFFE

'Client'

spiffe

spiffe

'Server'

Mutual TLS (transport layer security)

image credit: flaticon.com
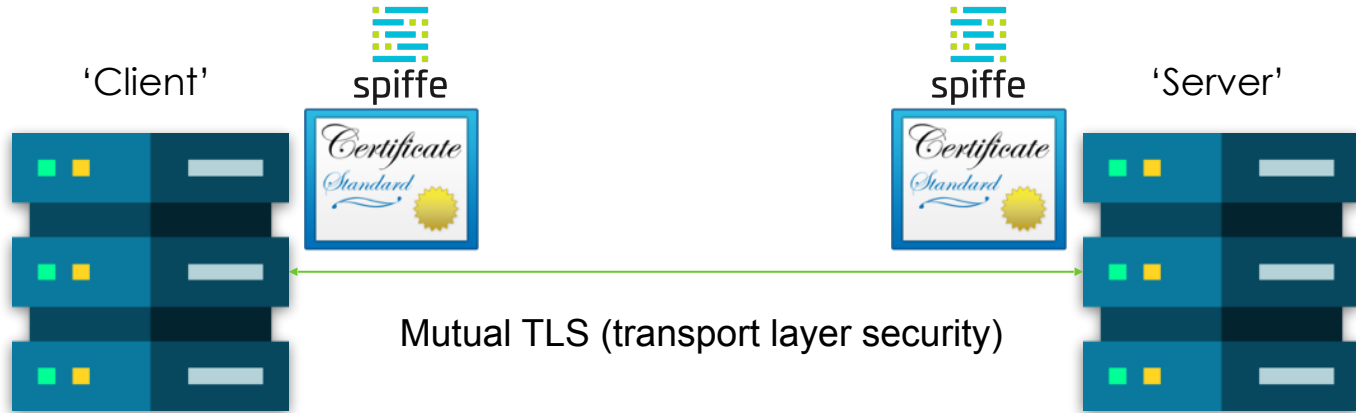
Meet SPIFFE

```
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
```
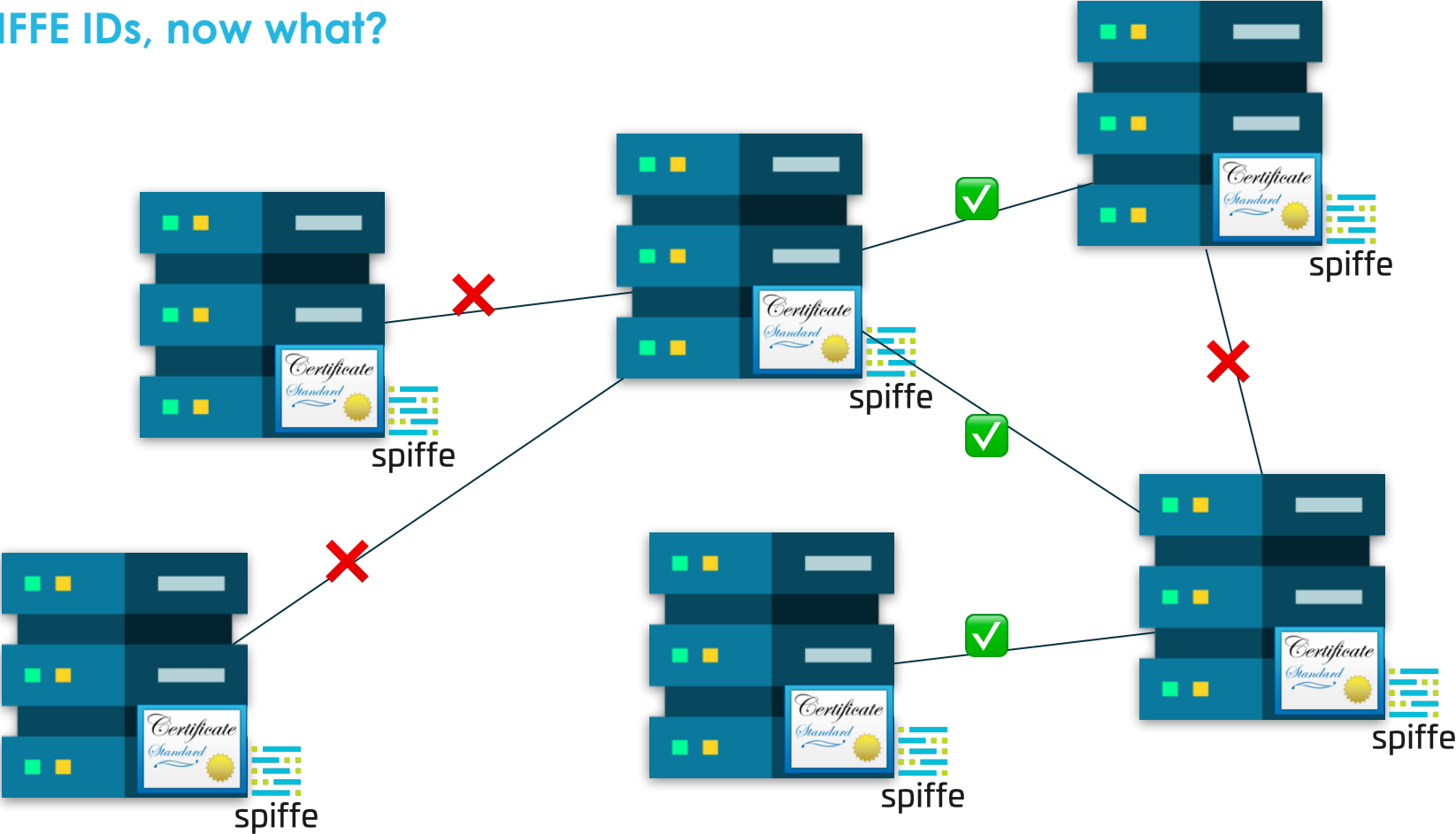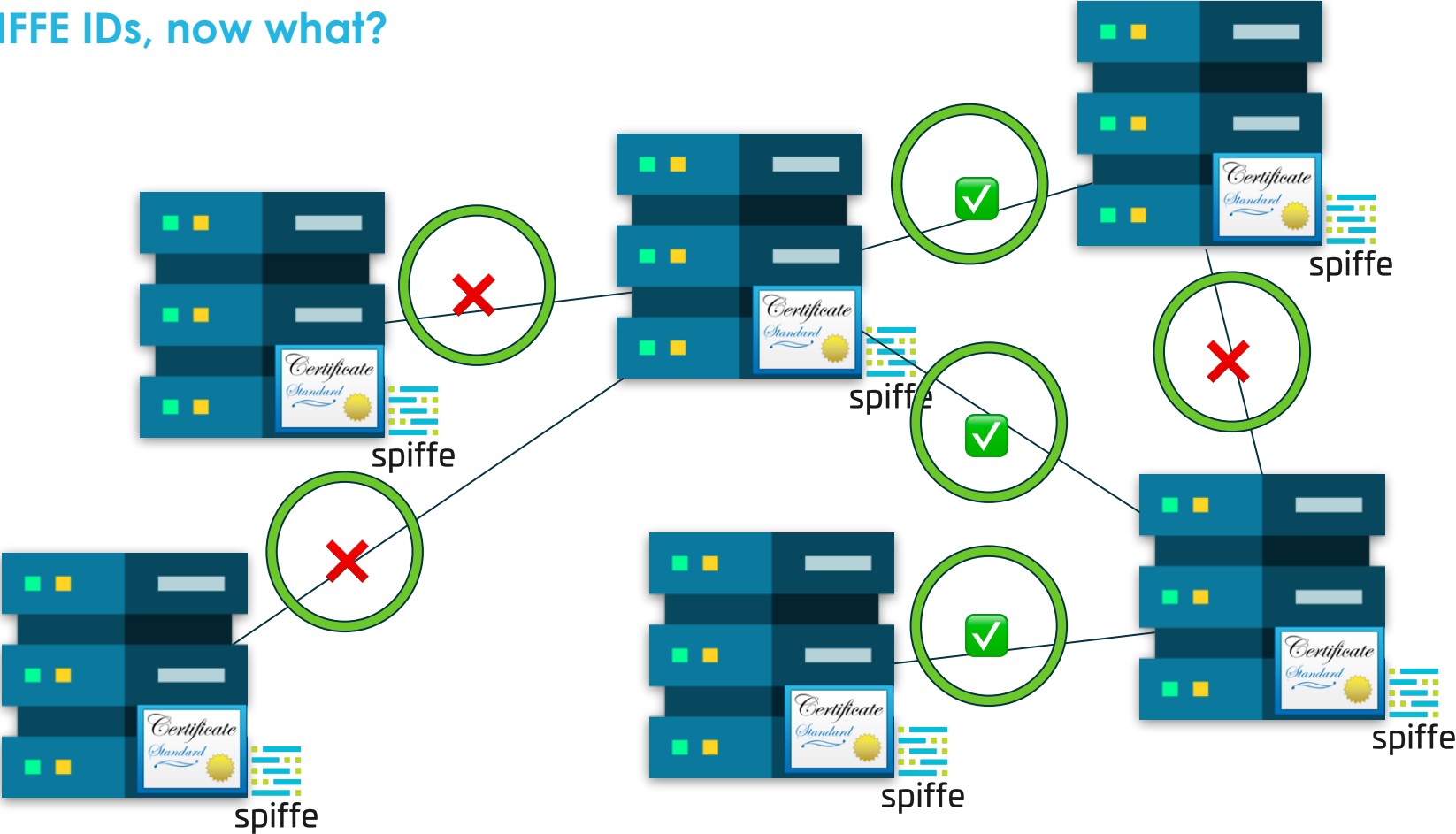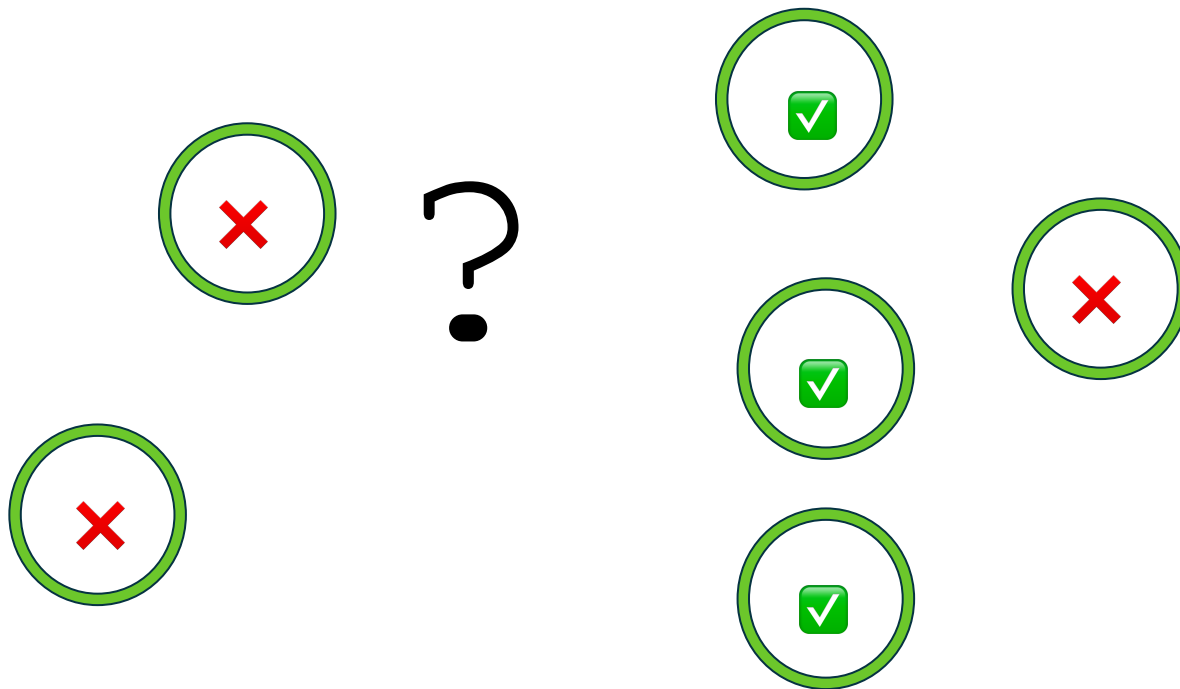
SPIFFE IDs, now what?

spiffe

spiffe

spiffe

spiffe

spiffe

spiffe

image credit: flaticon.com

SPIFFE IDs, now what?

spiffe

spiffe

spiffe

spiffe

spiffe

image credit: flaticon.com

SPIFFE IDs, now what?

image credit: flaticon.com
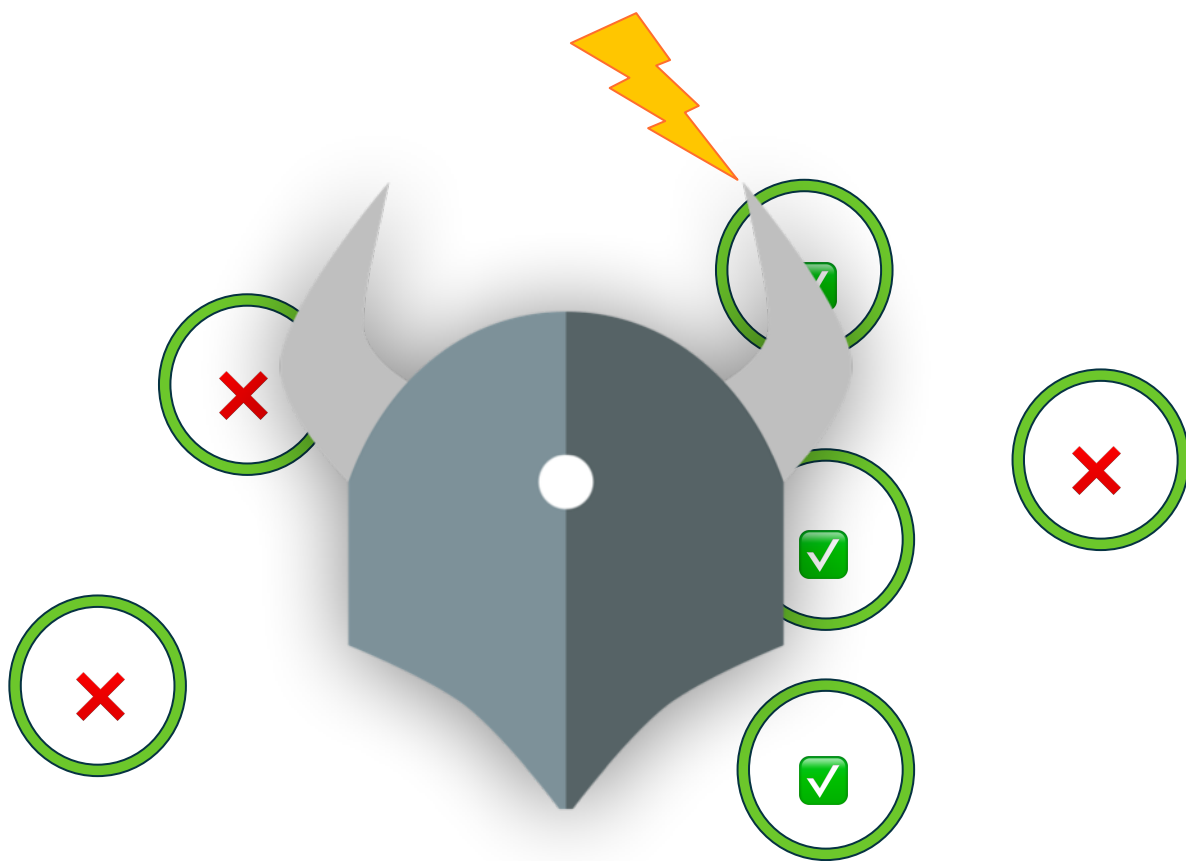
SPIFFE IDs, now what?

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
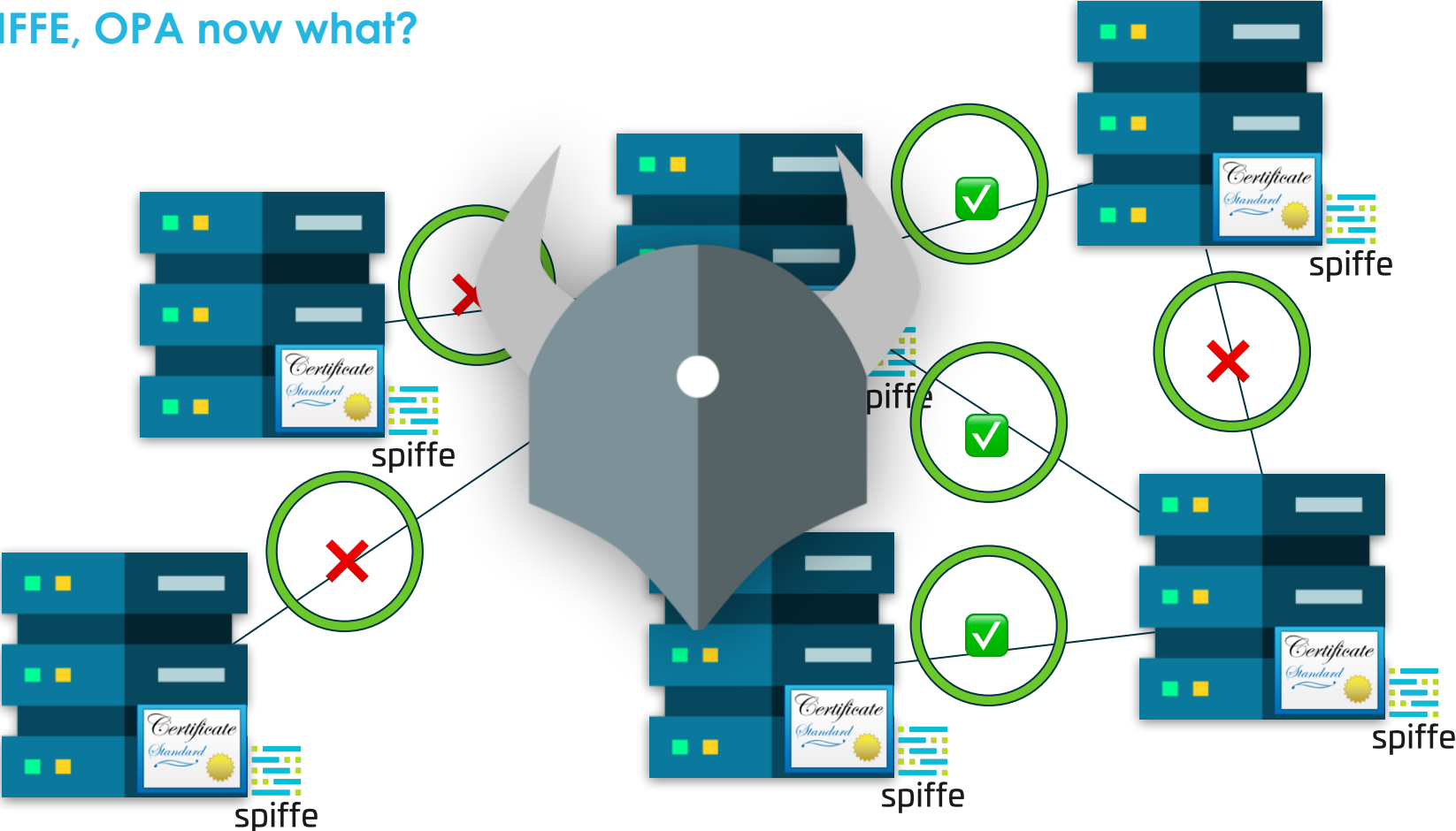
Example Request:

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

Example Response:

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

```rego
1  package authz
2
3  import future.keywords.if
4
5  default allow := false
6
7  allow if input.user.role == "admin"
```
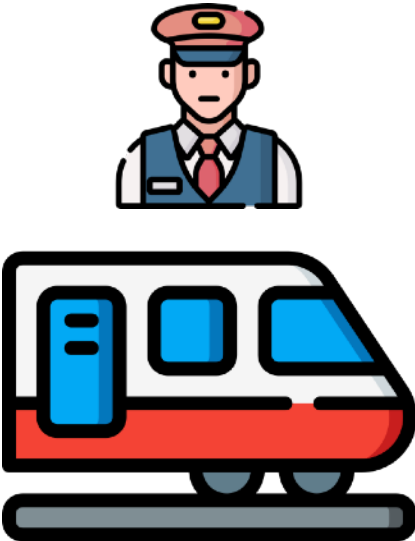
SPIFFE, OPA now what?

spiffe

spiffe

spiffe

spiffe

spiffe

Demo Environment Scenario

Demo Environment Scenario

Train Driver in Train

Local Station

HQ

image credit: flaticon.com

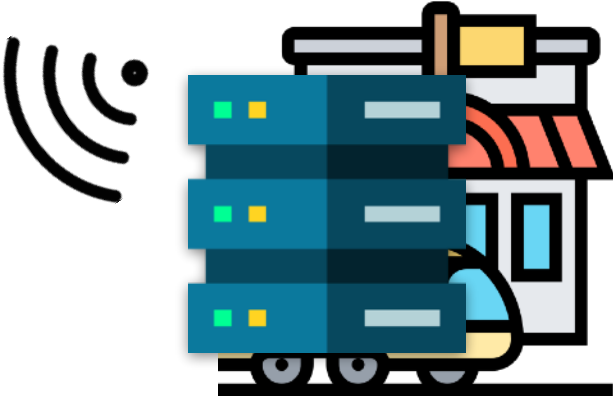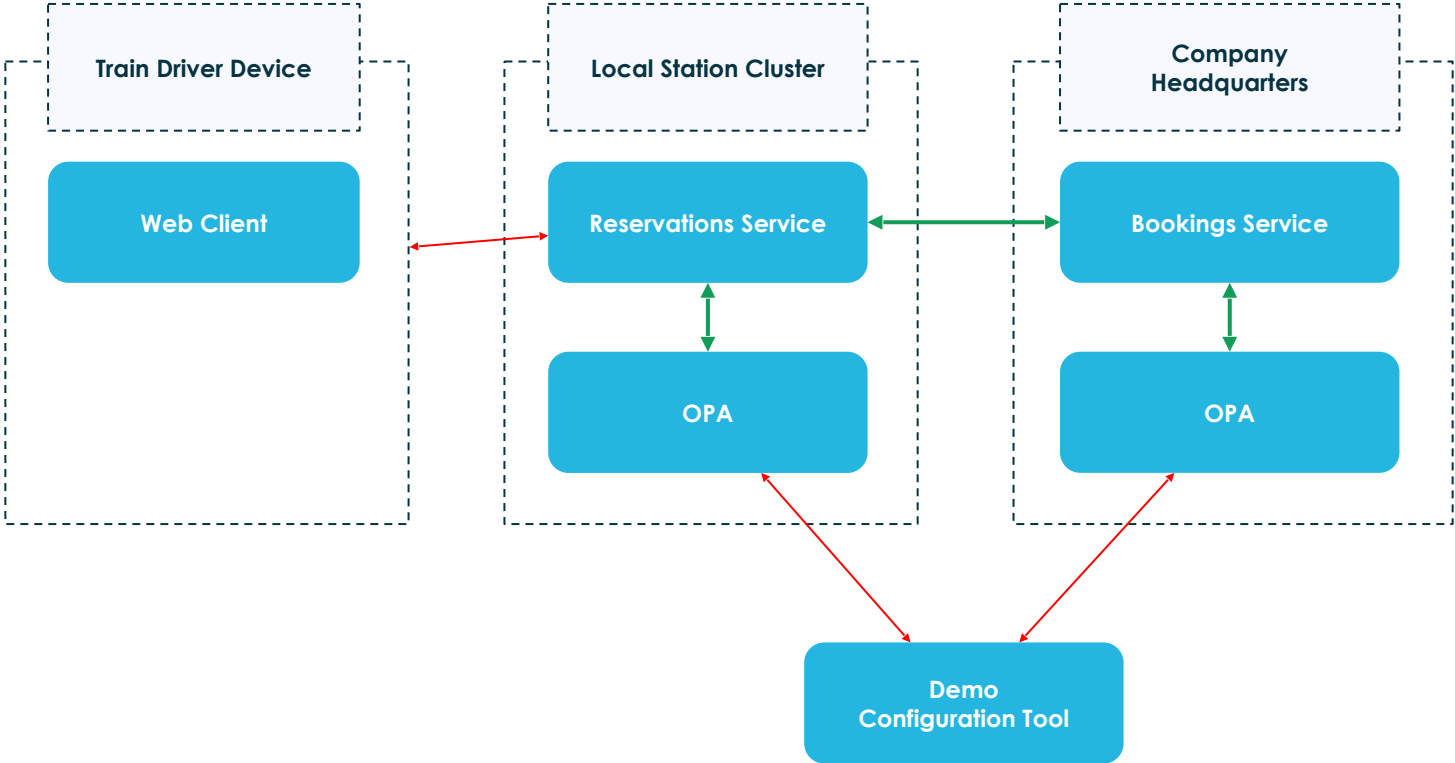# Demo Environment Scenario

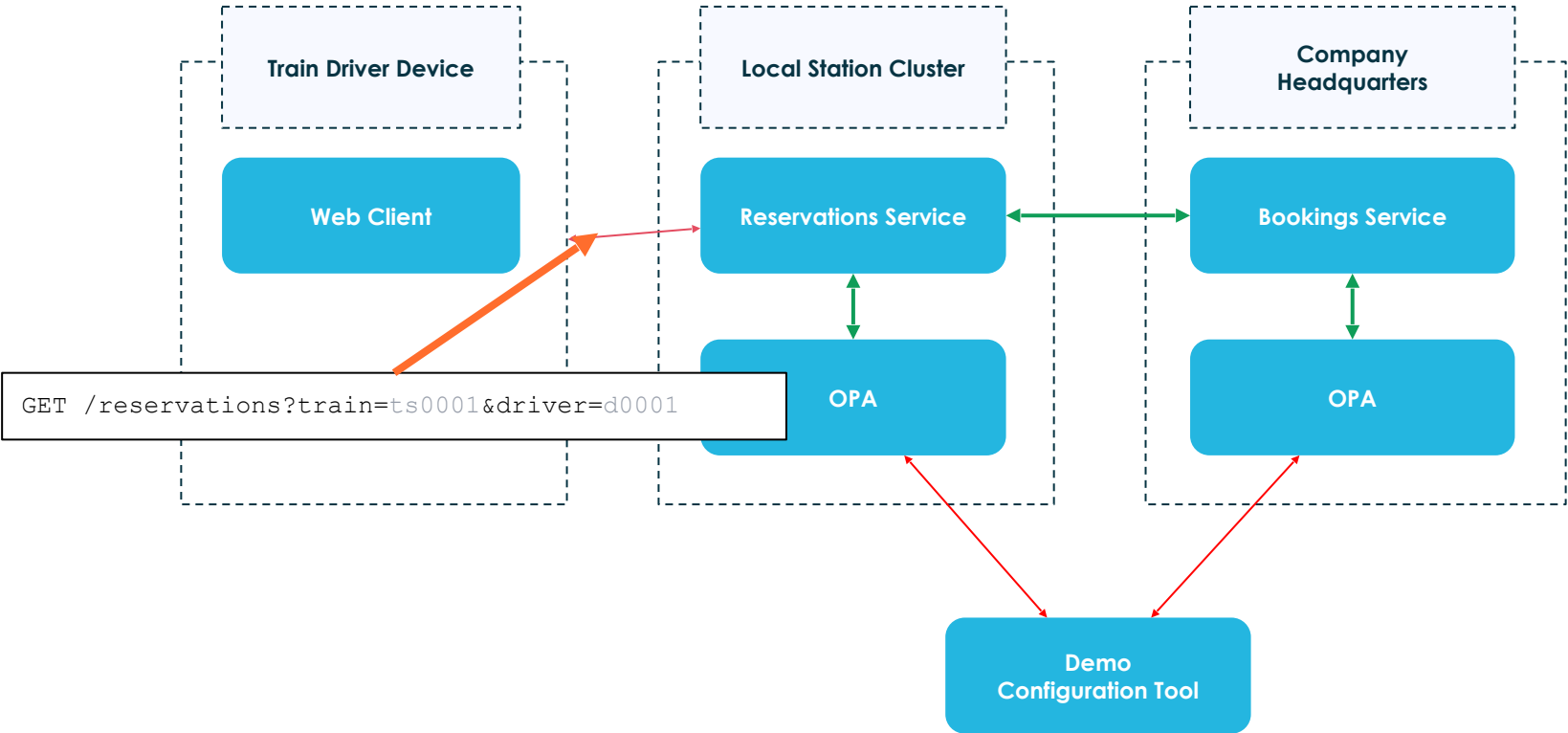Train Driver in Train

Local Station

HQ

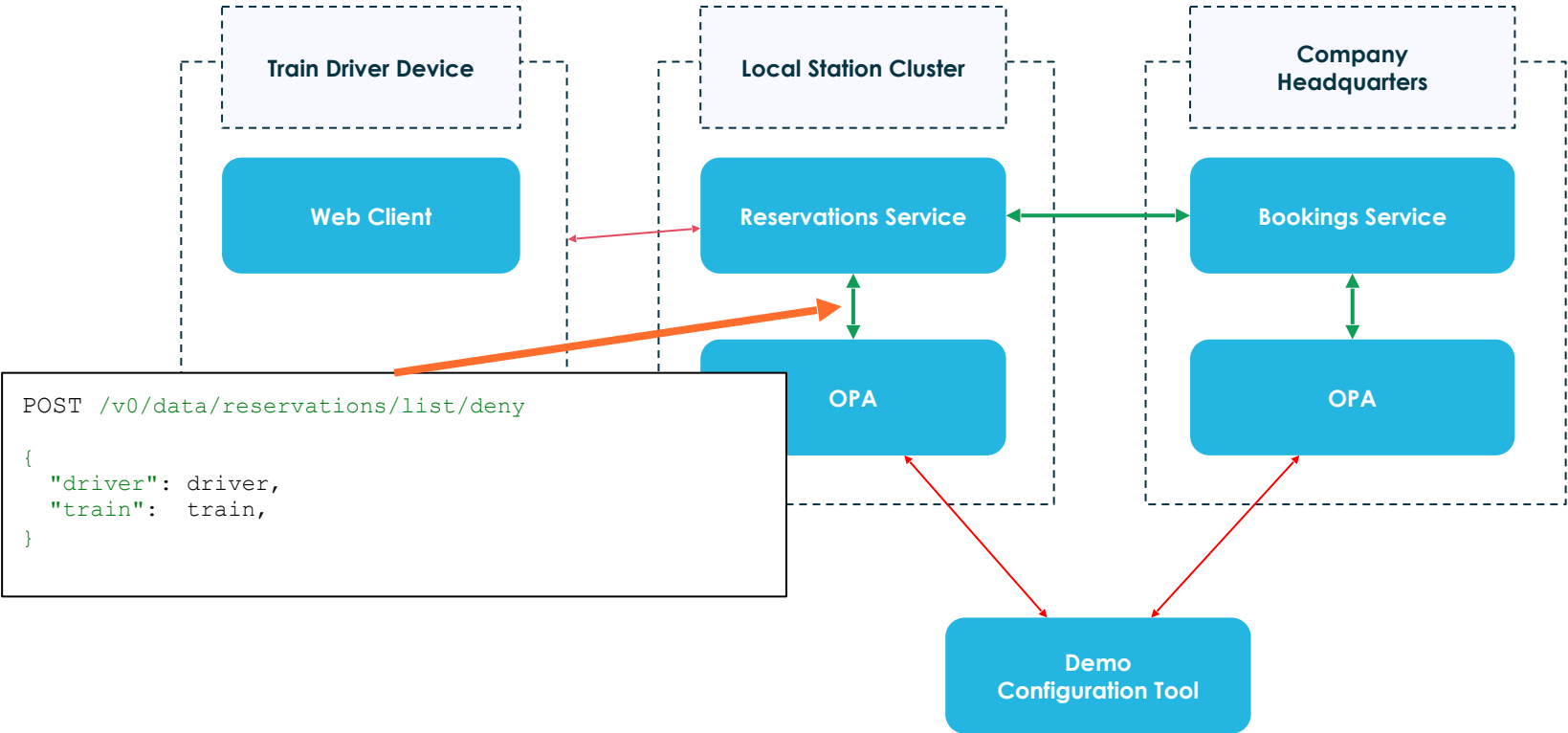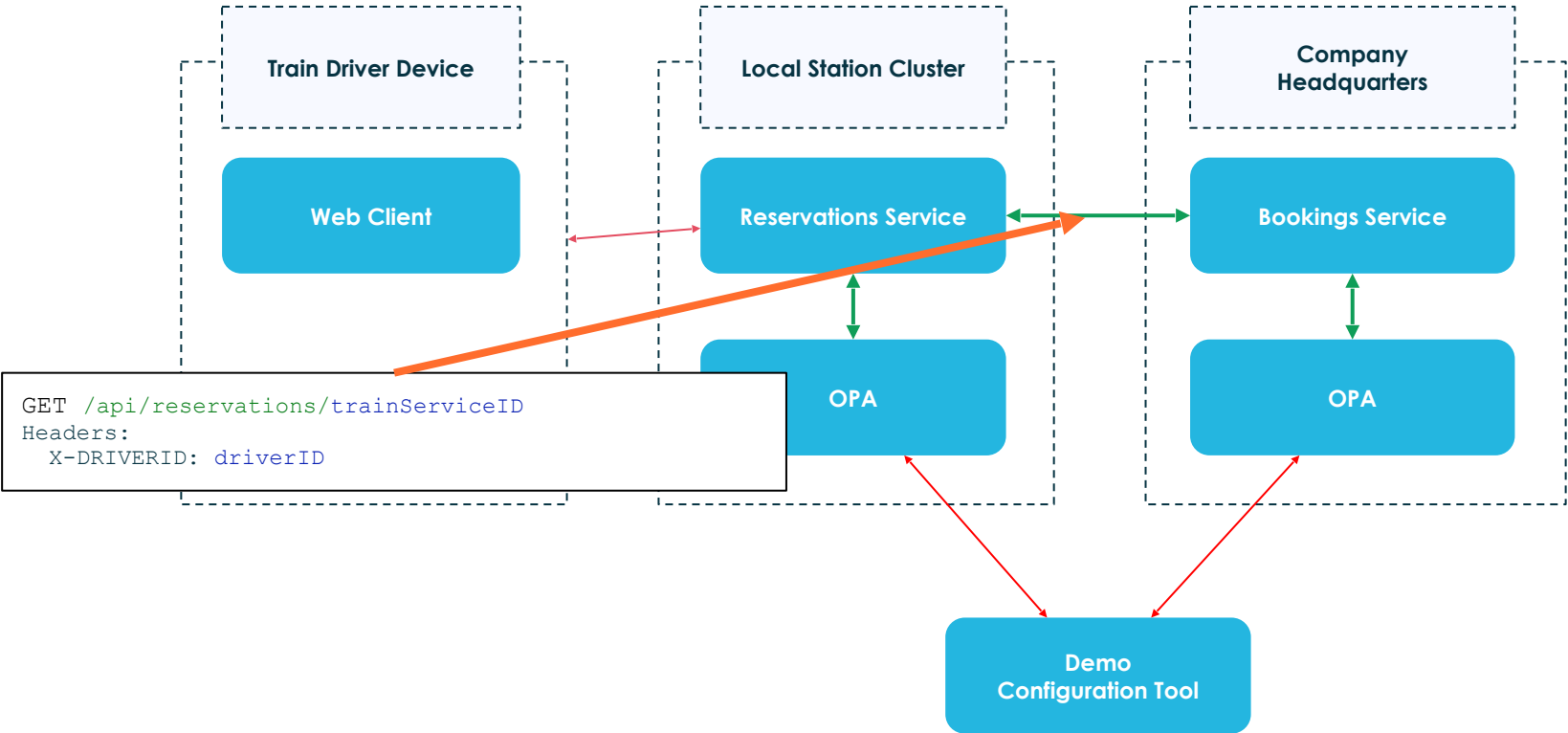image credit: flaticon.com

Demo Environment Architecture

Demo Environment Architecture

Demo Environment Architecture

Demo Environment Architecture

Demo Environment Architecture
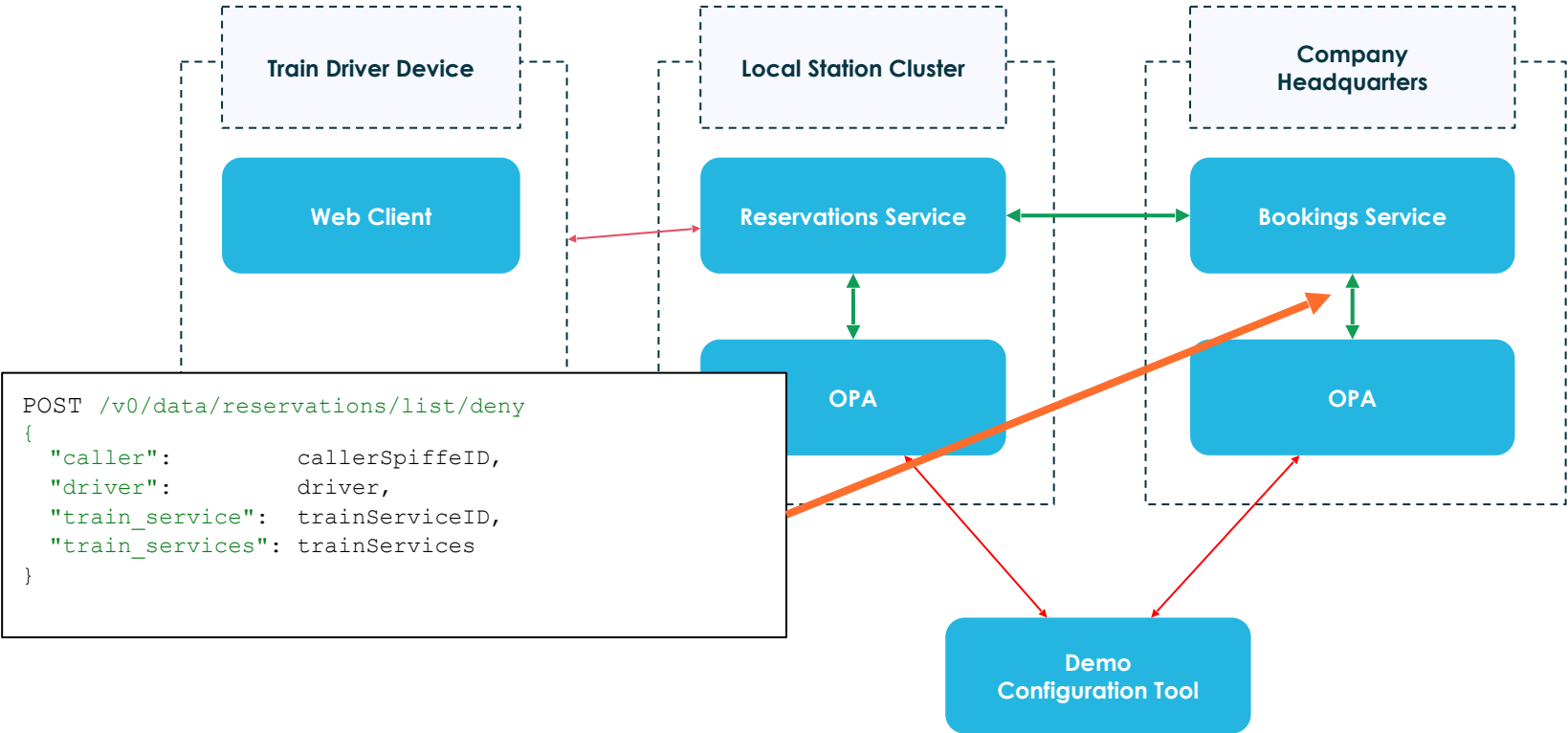


```
POST /v0/data/reservations/list/deny
{
  "caller":        callerSpiffeID,
  "driver":        driver,
  "train_service": trainServiceID,
  "train_services": trainServices
}
```

Demo Environment Architecture


```
POST /v0/data/reservations/list/option

{
  "caller": callerSpiffeID
}
```
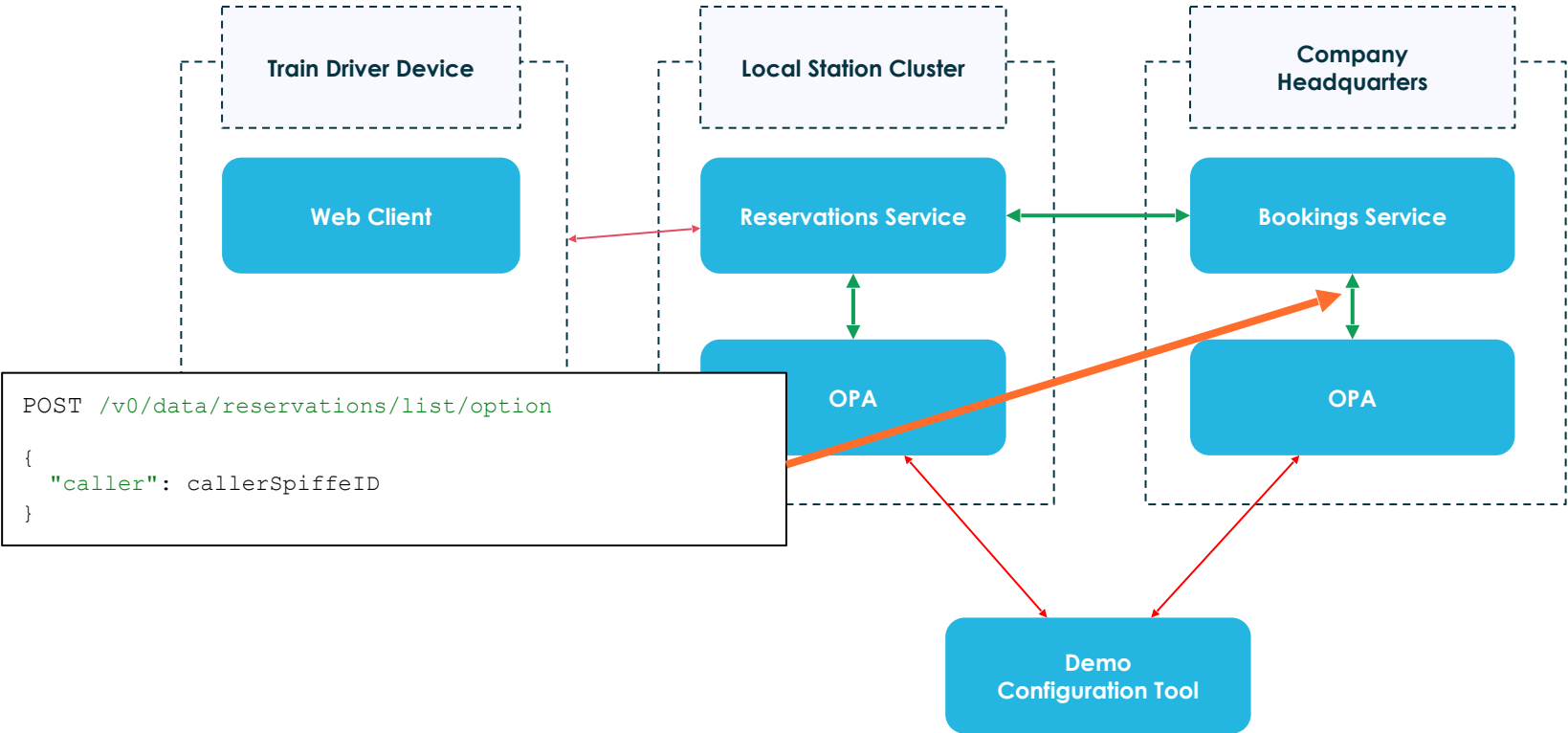
Demo Environment Architecture
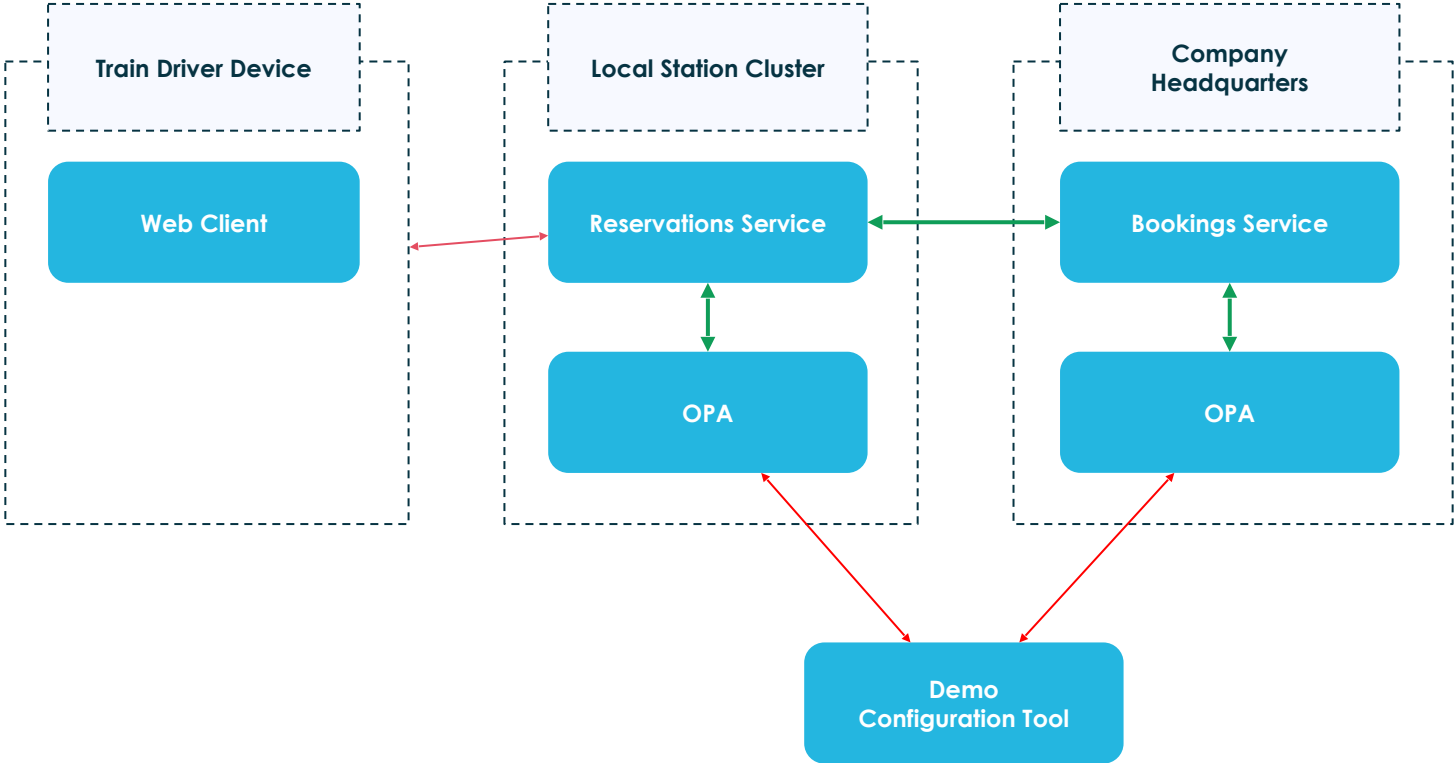
Demo Summary

In this demo we have seen:
- SPIFFE IDs used in OPA authorization policies
- OPA used to enforce access
- OPA used to control application behavior

Work done in OPA for this presentation

- Support authorizing callers with SPIFFE IDs to OPA
  - https://github.com/open-policy-agent/opa/pull/5742

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

- Styra Products
  - https://www.styra.com/styra-das/ (production version of my demo config tool)
  - https://www.styra.com/styra-load/
- Jetstack Blog:
  - https://www.jetstack.io/blog/workload-identity-with-spiffe-trust-domains/
- Code: https://github.com/charlieegan3/talk-opa-spiffe
- Slides: https://charlieegan3.com/talks/opa-spiffe/slides.pdf
- Links: https://charlieegan3.com/talks/opa-spiffe/

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

Thank You

styra