KubeCon | CloudNativeCon

Europe 2018

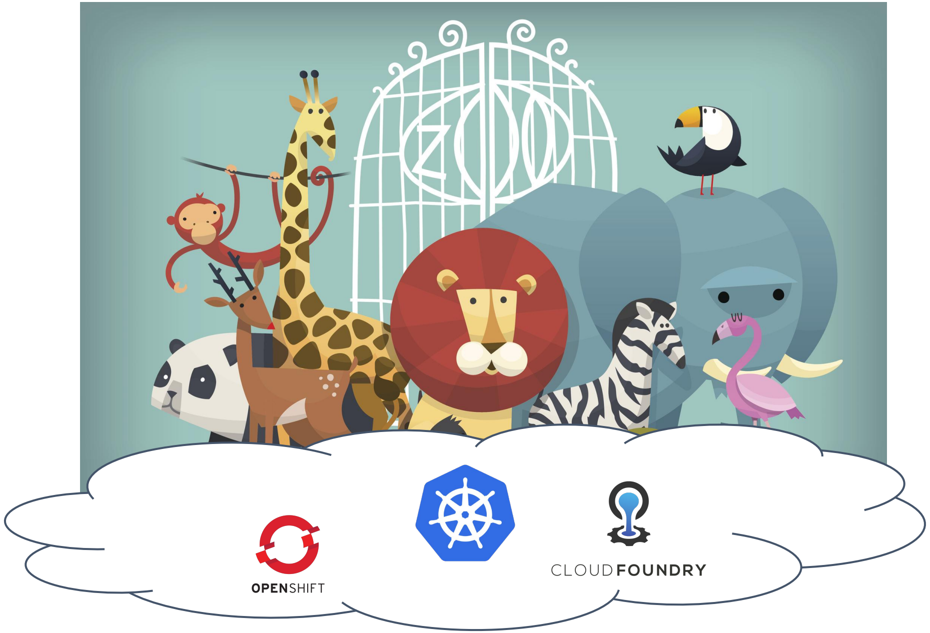
Cloud Native Identity Management

Andreas Zitzelsberger, QAware
Andrew Jessup, Scytale.io

Once upon a time...

Large cloud project for a major company

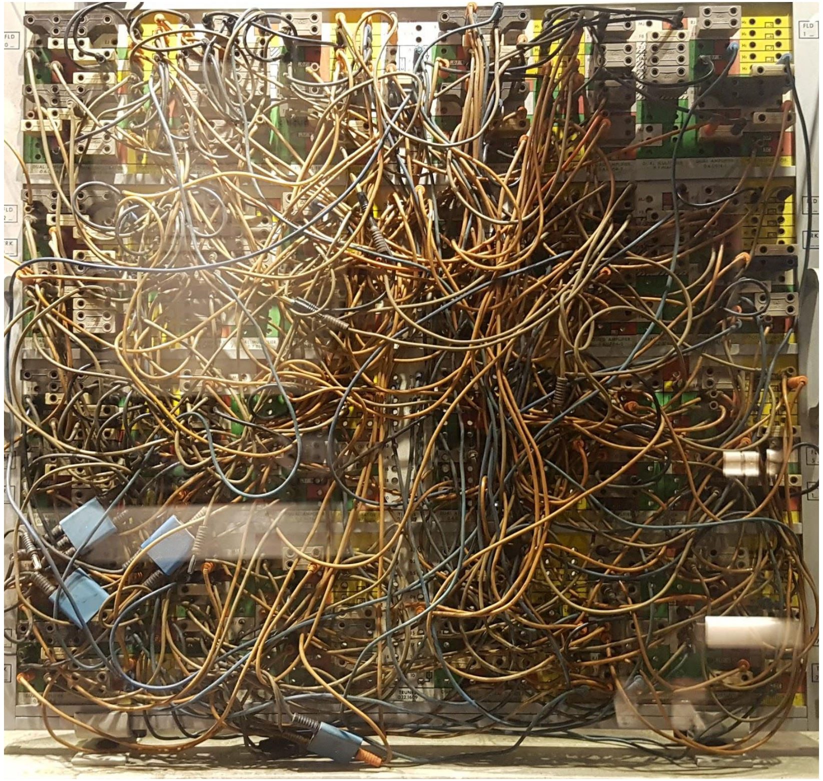
- Hundreds of apps in the cloud
- Many more on-prem
- Little centralized control
- Strict legal requirements
- Strict security requirements



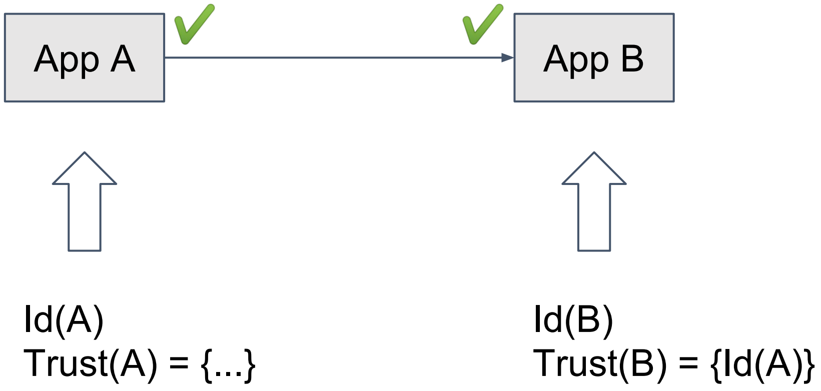
The Good, the Bad and the Ugly of Migrating Hundreds of Legacy Applications to Kubernetes, Josef Adersberger, KubeCon 2017
https://bit.ly/2JZNRHw

Where did we start?

- Classic approach:
  0-trust with TLS / X.509
- Secure
- But: Decomposition of applications leads to an explosion of trust relations
  - Hard to manage at scale
  - Complex and error-prone
- Also, no secret rotation



Let’s take a step back and look at the problem...



App A ✓

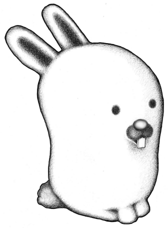
✓ App B

Id(A)
Trust(A) = {...}

Id(B)
Trust(B) = {Id(A)}

- Secure Authentication and Authorization
- Scale
- Dynamicity
- Manageability
- Secret rotation
- Interoperability
- Hybrid cloud

spiffe



11th USENIX Security Symposium (2002)
Plan9 security design published

Circa 2005
Google develops the Low Overhead Authentication Service

GlueCon 2016
Joe Beda proposes SPIFFE

KubeCon NA 2017
SPIFFE & SPIRE 0.1 are released

spiffe

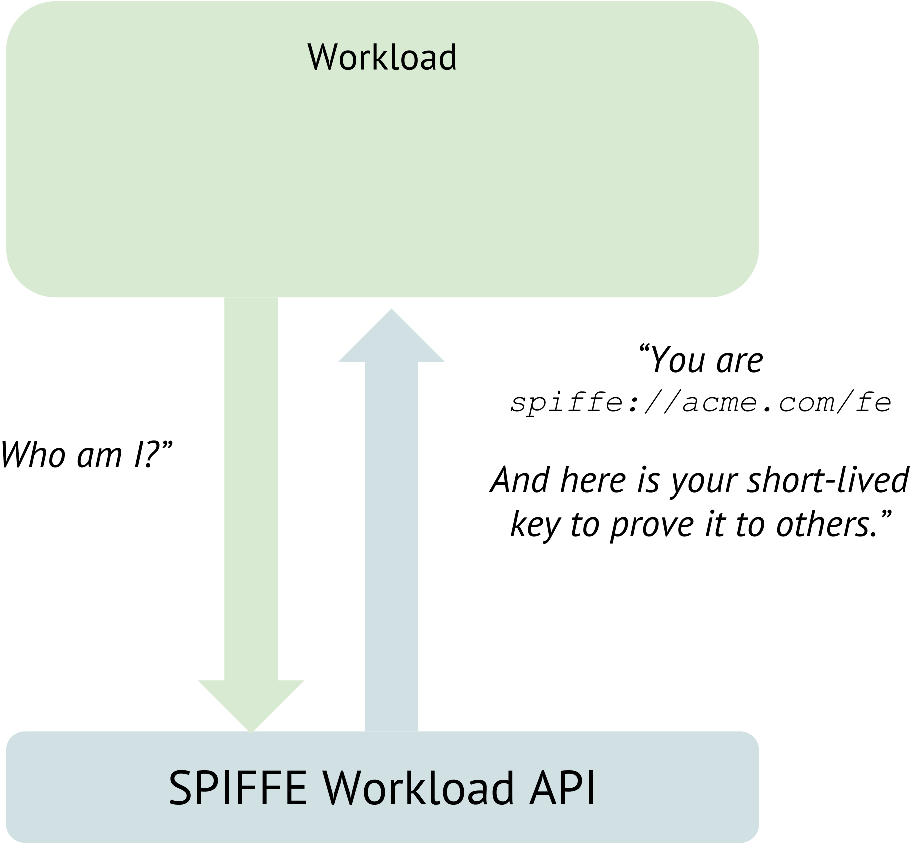
April 2018
SPIFFE & SPIRE accepted into the CNCF

spiffe

SCYTALE

TIGERA

heptio



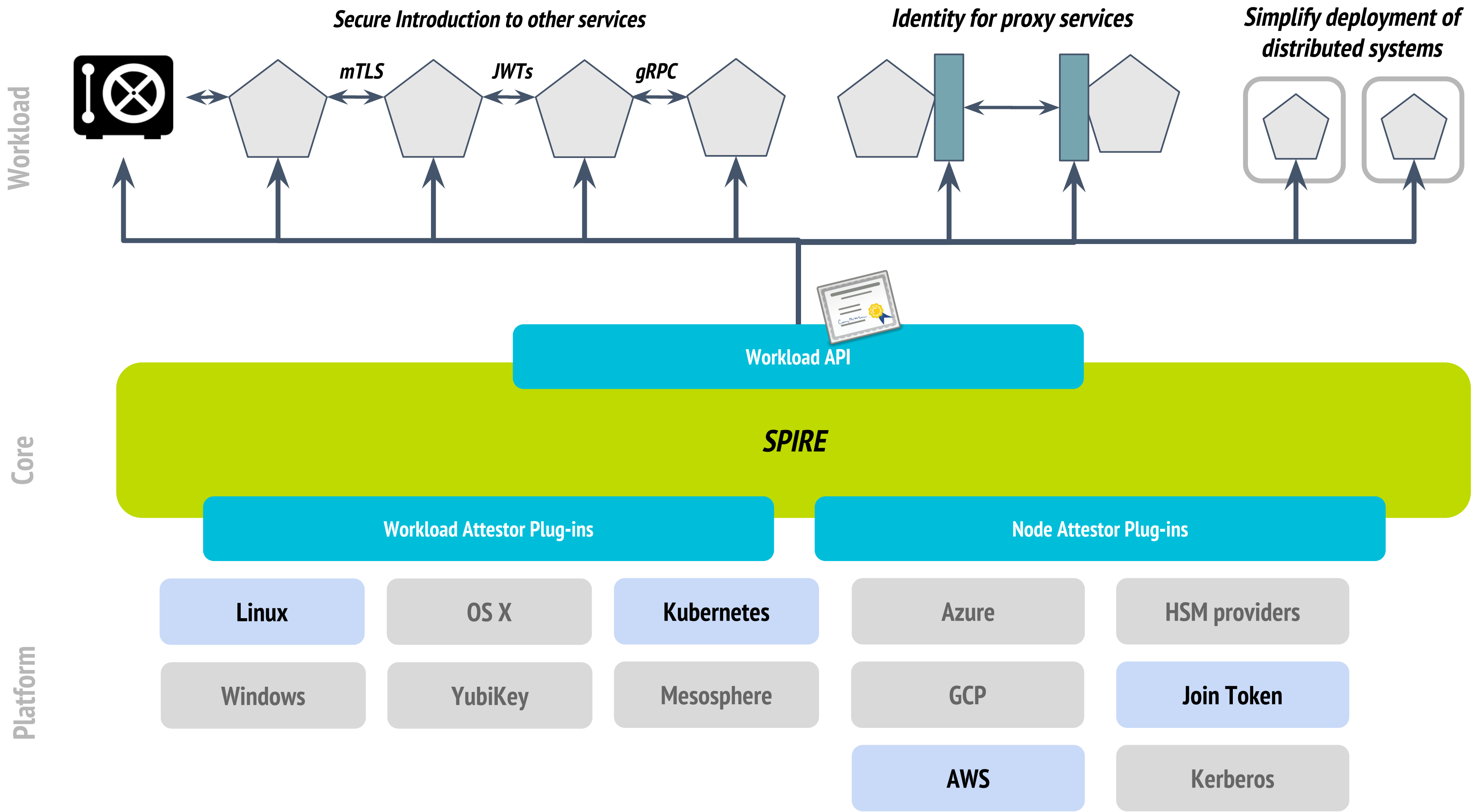
Workload

"Who am I?"

"You are `spiffe://acme.com/fe`

And here is your short-lived key to prove it to others."

SPIFFE Workload API



Workload

Core

Platform

Secure Introduction to other services

mTLS

JWTs

gRPC

Identity for proxy services

Simplify deployment of distributed systems

Workload API

SPIRE

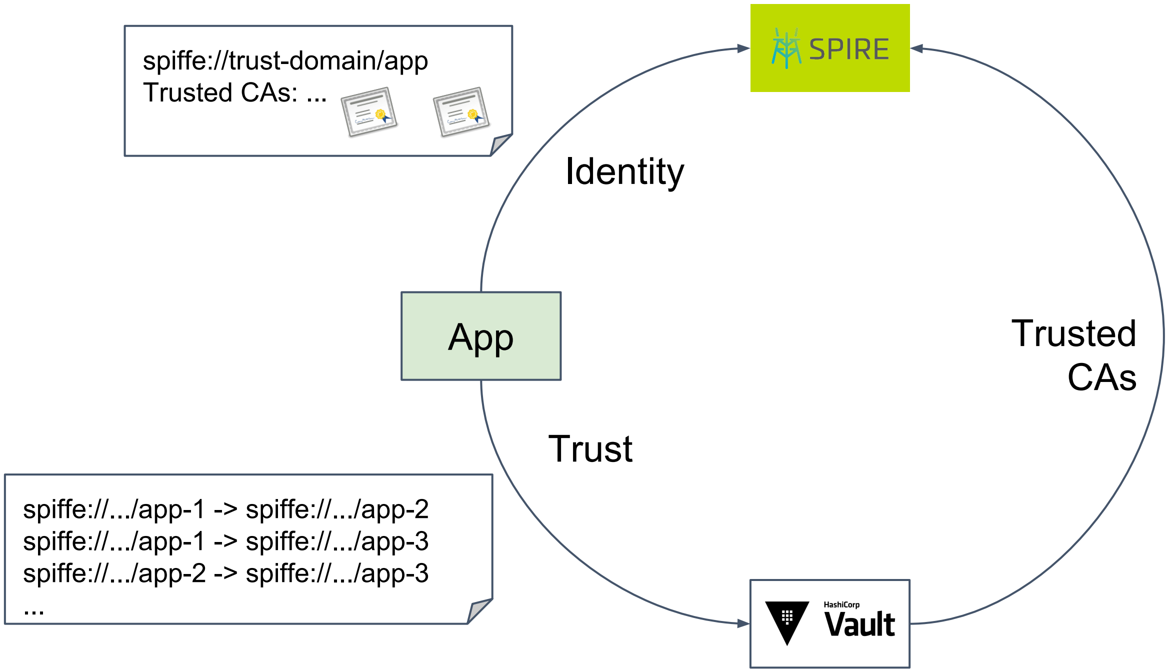
Workload Attestor Plug-ins

Node Attestor Plug-ins

Linux	OS X	**Kubernetes**	Azure	HSM providers
Windows	YubiKey	Mesosphere	GCP	**Join Token**
			AWS	Kerberos

Building on top of SPIFFE and SPIRE

SPIRE provides identity, Vault trust



spiffe://trust-domain/app
Trusted CAs: ...

spiffe://.../app-1 -> spiffe://.../app-2
spiffe://.../app-1 -> spiffe://.../app-3
spiffe://.../app-2 -> spiffe://.../app-3
...

+ Rotating credentials for Databases, RabbitMQ, …
+ Secrets

Proper secret rotation is surprisingly hard

- Assumption: Keys and certificates are (a) static and (b) provided via files
  - Python (with Flask)

    ```
    app.run(ssl_context=('cert.pem', 'key.pem'))
    ```
  - Go (GRPC with TLS)

    ```
    credentials.NewServerTLSFromFile(crt, key)
    ```
  - Also Envoy, Nginx, …
- Java

  ```
  -Djavax.net.ssl.trustStore=... -Djavax.net.ssl.keyStore=...
  ```
  - But Java has the java.security API
  - Certificates and keys can be rotated online (if the API is used properly)

Integrating Vault and SPIRE

Transport (TLS) Authentication

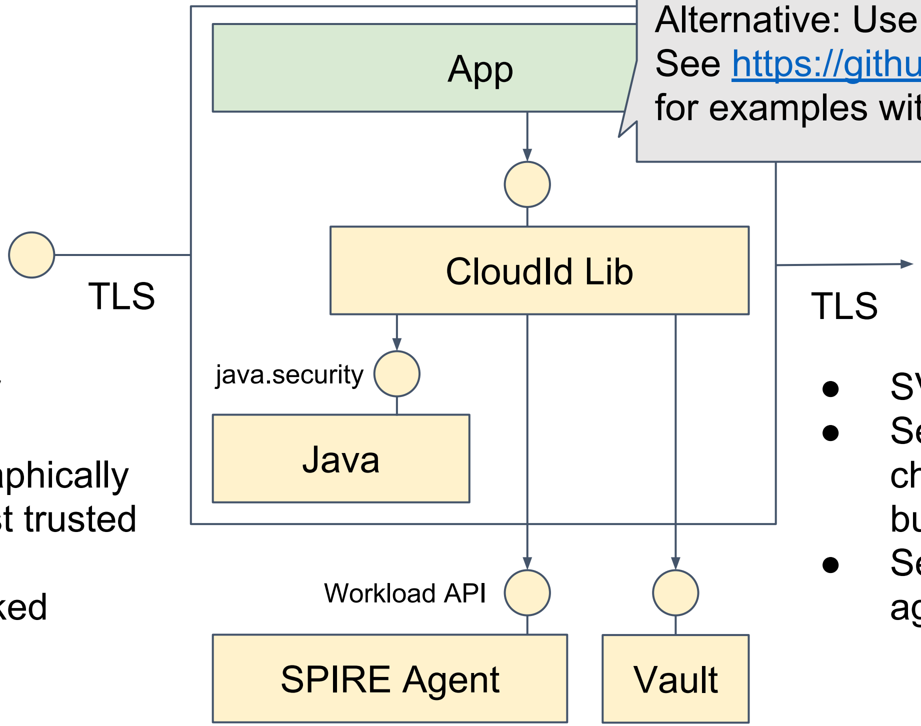
- Unfriendly to certificate rotation due to unseal process
- *Solution*: Put Vault alongside SPIRE in the same PKI

App Authentication

- Previously unable to validate URI SANs because Go up to 1.9 lacked support
- Works from Vault 0.10.2 on (PR #4231)
- *Solution*: Sidecar regularly updates the trusted auth certificate with the SPIRE CA

Piecing it all together

Integration at application level.
Alternative: Use an ambassador proxy
See https://github.com/spiffe/spiffe-example
for examples with Envoy and Ghosttunnel



- SVID as Server Certificate
- Client cryptographically checked against trusted bundles
- Client Ids checked against ACL

- SVID as Client Certificate
- Servers cryptographically checked against trusted bundles
- Server Ids checked against ACL

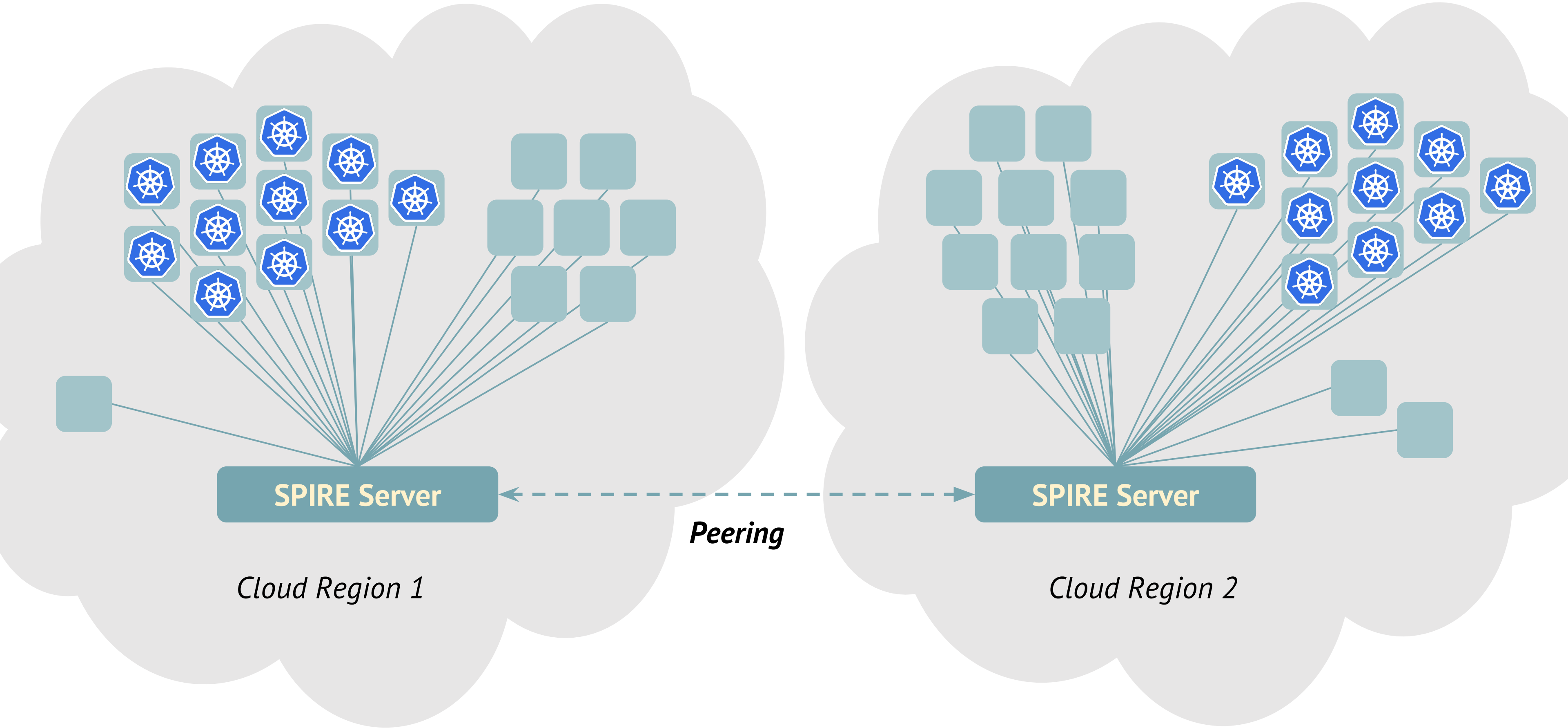
Demo time

https://github.com/qaware/cloudid-showcase

Our next steps...

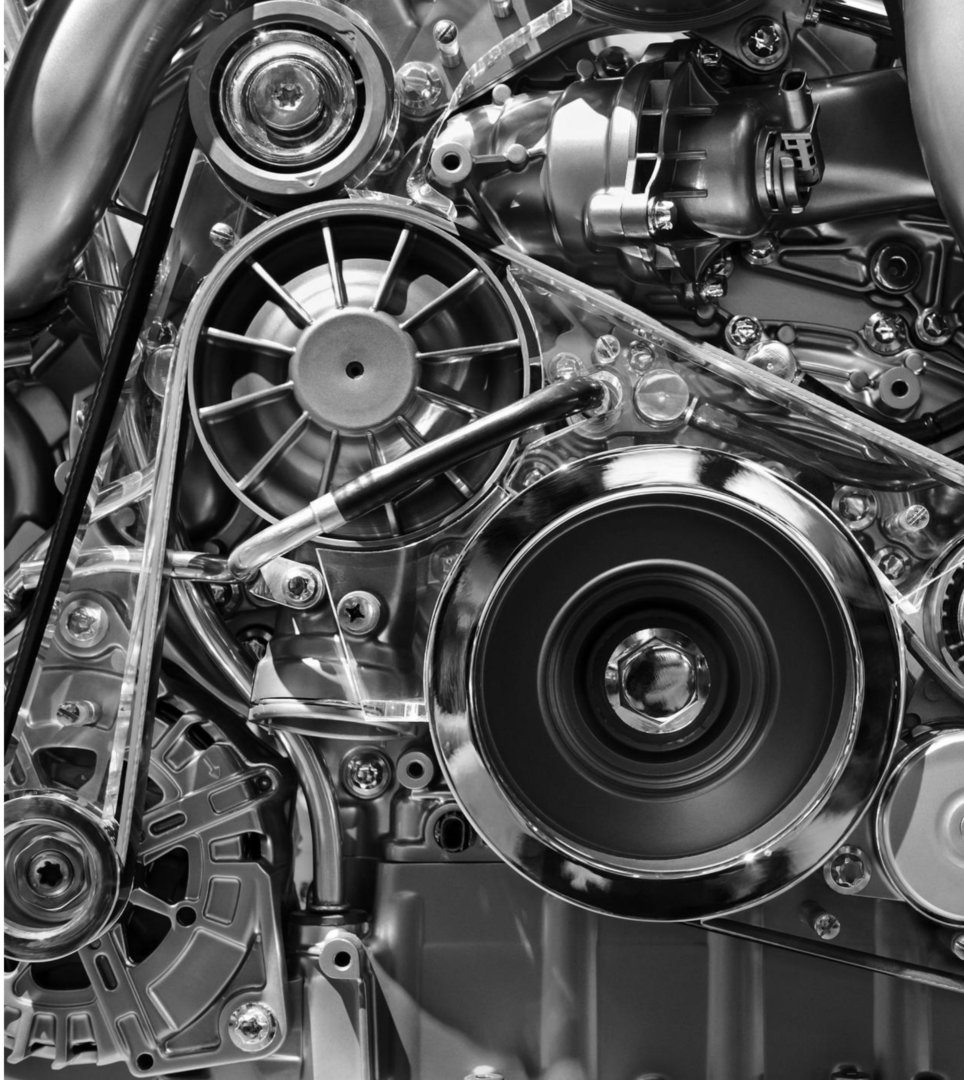
Further exploration

- Use the SPIFFE Id for tracing and correlating logs
- Interaction with other service meshes
- Connect workload and user identities
- Federation and hybrid cloud


SPIRE Server
Peering
SPIRE Server
Cloud Region 1
Cloud Region 2

Summary

There’s a lot of infrastructure...



… exposed with a single line of code.

The configuration effort is reduced to the actual business problem

- Specify who is who
- Specify who is allowed to talk to whom

spiffe SPIRE

May 2 (Today)	**May 3 (Tomorrow)**	**May 4 (Friday)**
	TheNewStack Pancake Breakfast talks SPIFFE *7.30am*	
SPIFFE Project Intro 4.25pm	SPIFFE Deep Dive (*Scytale*) *2pm*	Panel: App Security Requires Containers *4.25pm*

spiffe.io | github.com/spiffe | slack.spiffe.io

Thank You!

Andreas Zitzelsberger

@andreasz82

Andrew Jessup

@whenfalse

Special thanks to

Christian Fritz	QAware
Roman Buchholz	QAware
Evan Gilman	Syctale.io
Nic Jackson	Hashicorp