Five things you didn't know you could do with SPIFFE and SPIRE

Andrew Jessup and Andrés Vega

We're from Scytale. We work on SPIFFE and SPIRE.

Andrés

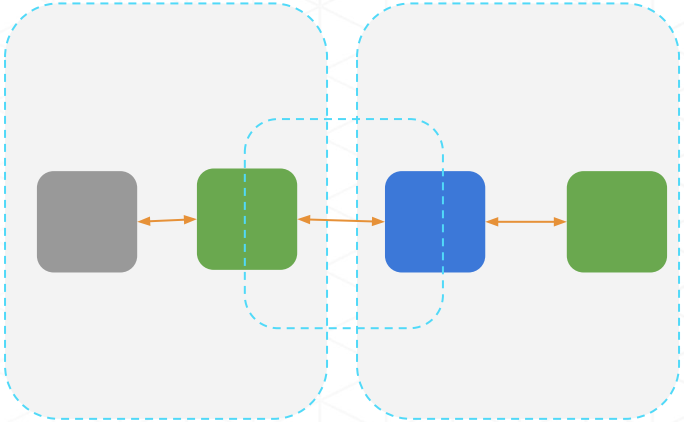
@invariantly

Andrew

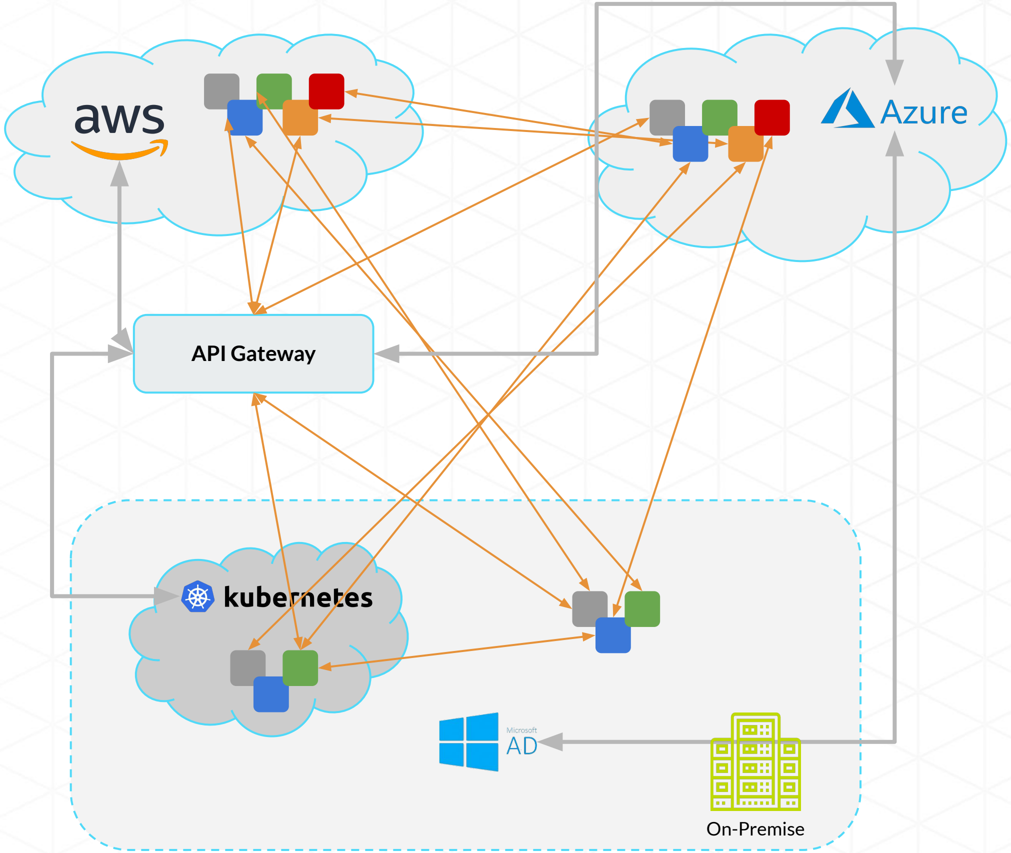
@whenfalse

Who are we?

Why **SPIFFE** now?



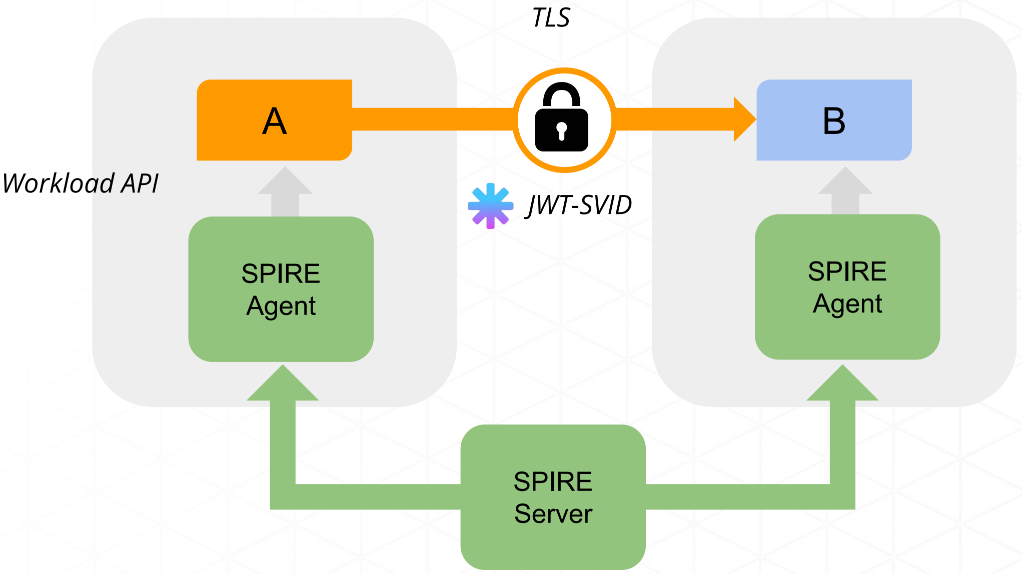
Why **SPIFFE** now?



KubeCon NA 2017
Austin, Texas

KubeCon NA 2018
Seattle, Washington

KubeCon NA 2019
San Diego, California



Define a standard ....and a toolchain

KubeCon NA 2017
Austin, Texas

KubeCon NA 2018
Seattle, Washington

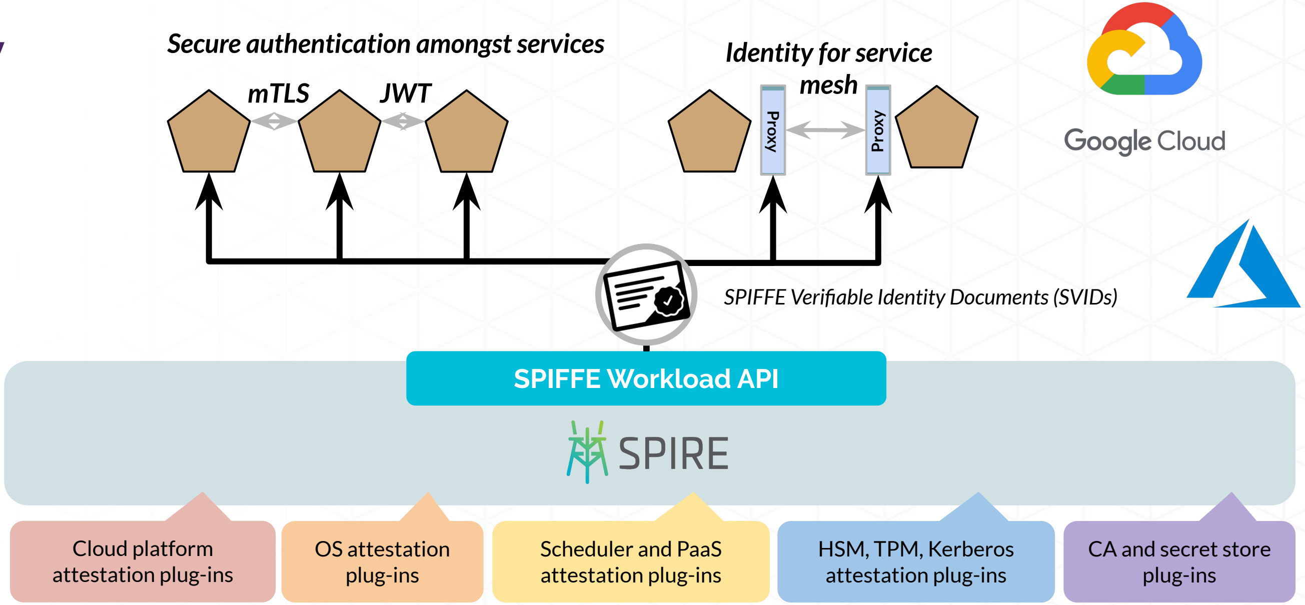
KubeCon NA 2019
San Diego, California

envoy

amazon web services

Google Cloud

Azure



Secure authentication amongst services

mTLS

JWT

Identity for service mesh

Proxy

Proxy

SPIFFE Verifiable Identity Documents (SVIDs)

SPIFFE Workload API

SPIRE

Cloud platform attestation plug-ins

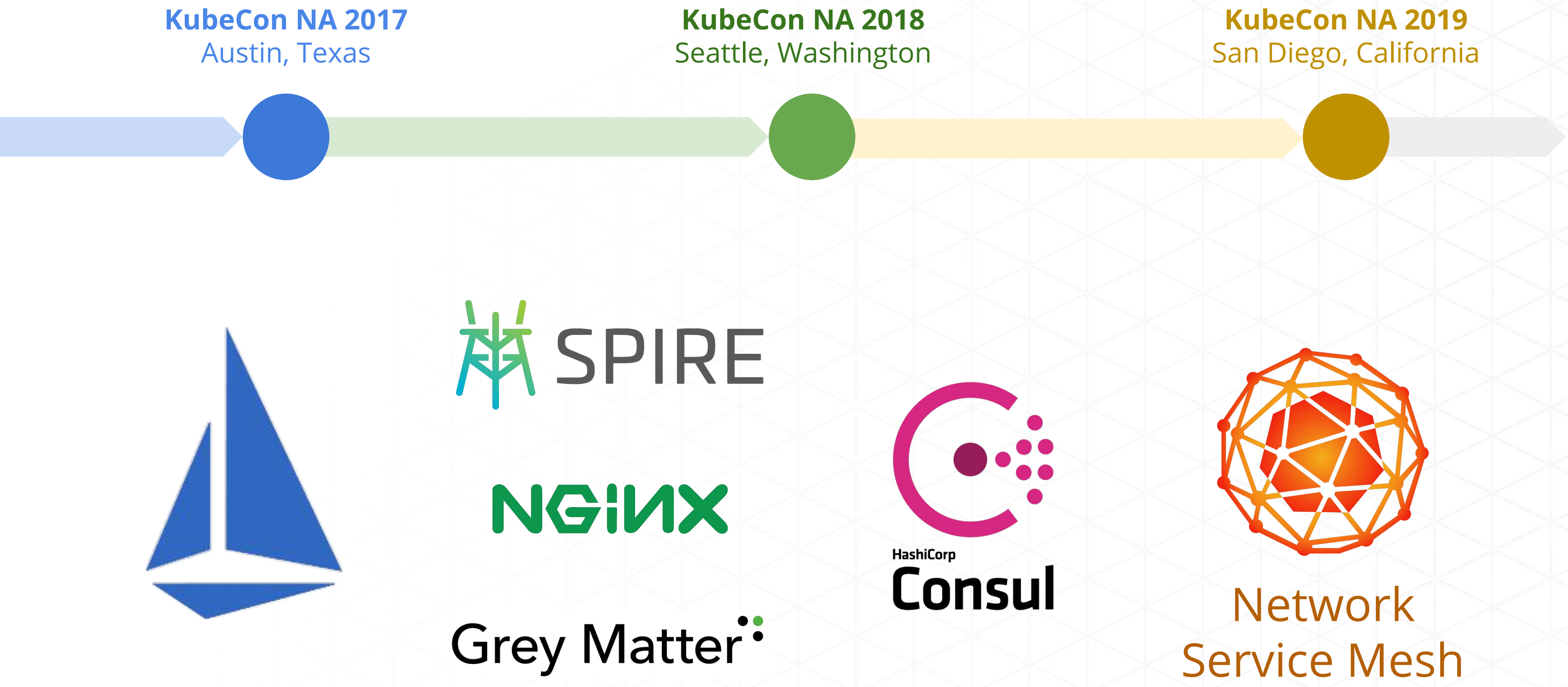
OS attestation plug-ins

Scheduler and PaaS attestation plug-ins

HSM, TPM, Kerberos attestation plug-ins

CA and secret store plug-ins

Solve for workload-to-workload communication



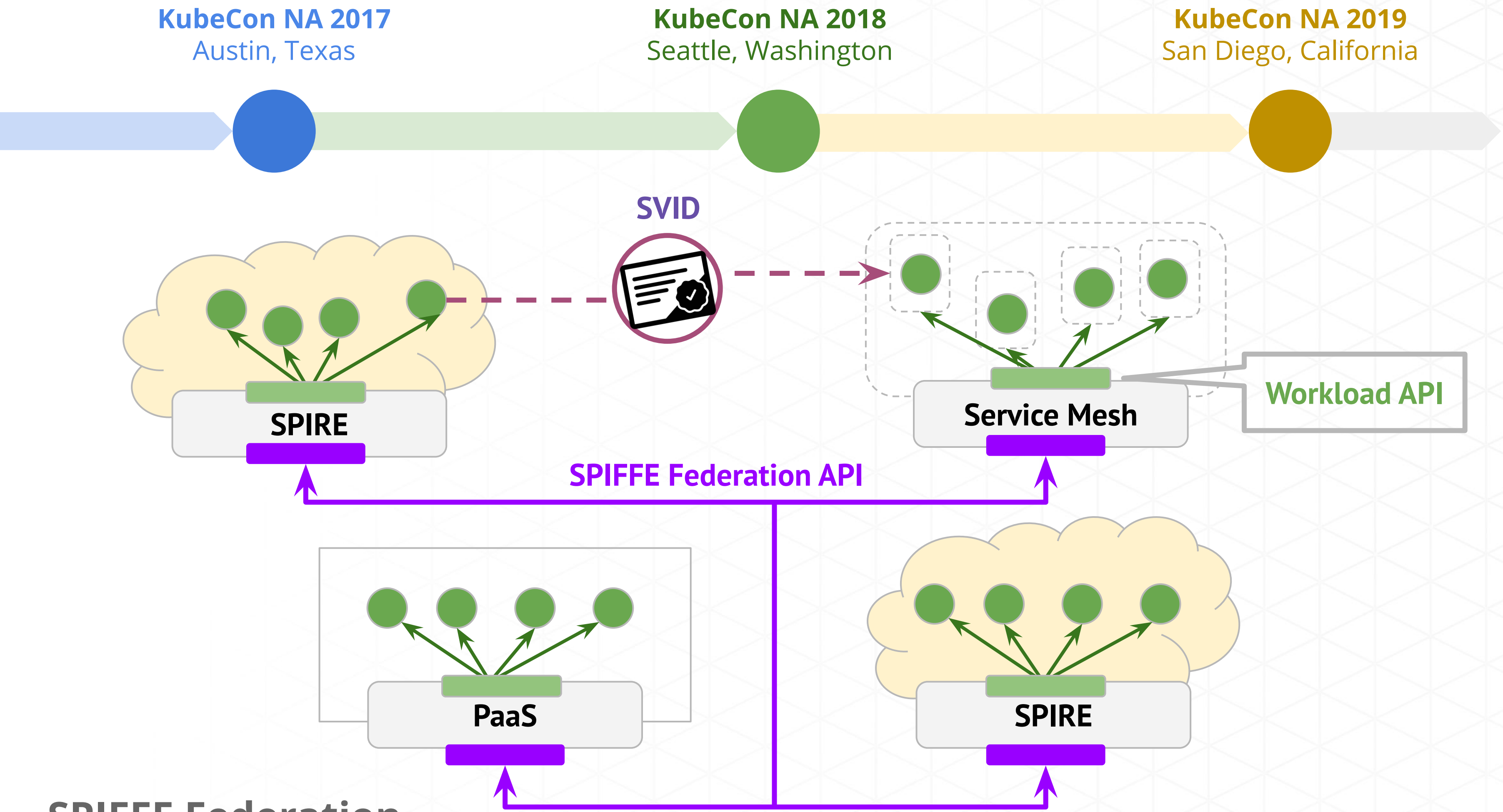
KubeCon NA 2017
Austin, Texas

KubeCon NA 2018
Seattle, Washington

KubeCon NA 2019
San Diego, California

Network
Service Mesh

SPIFFE has become bigger than SPIRE

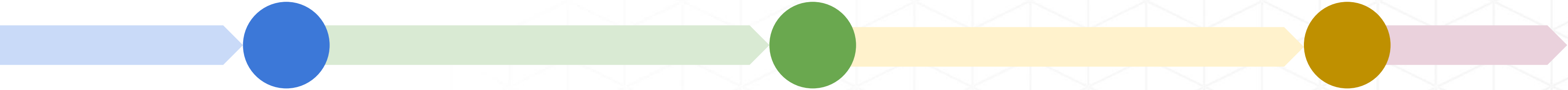


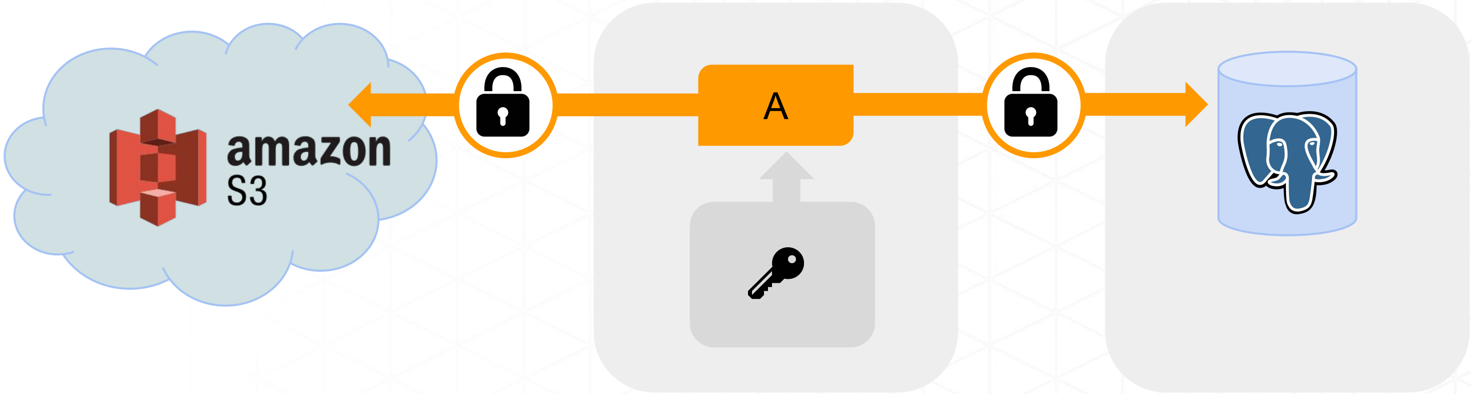
SPIFFE Federation

KubeCon NA 2017
Austin, Texas

KubeCon NA 2018
Seattle, Washington

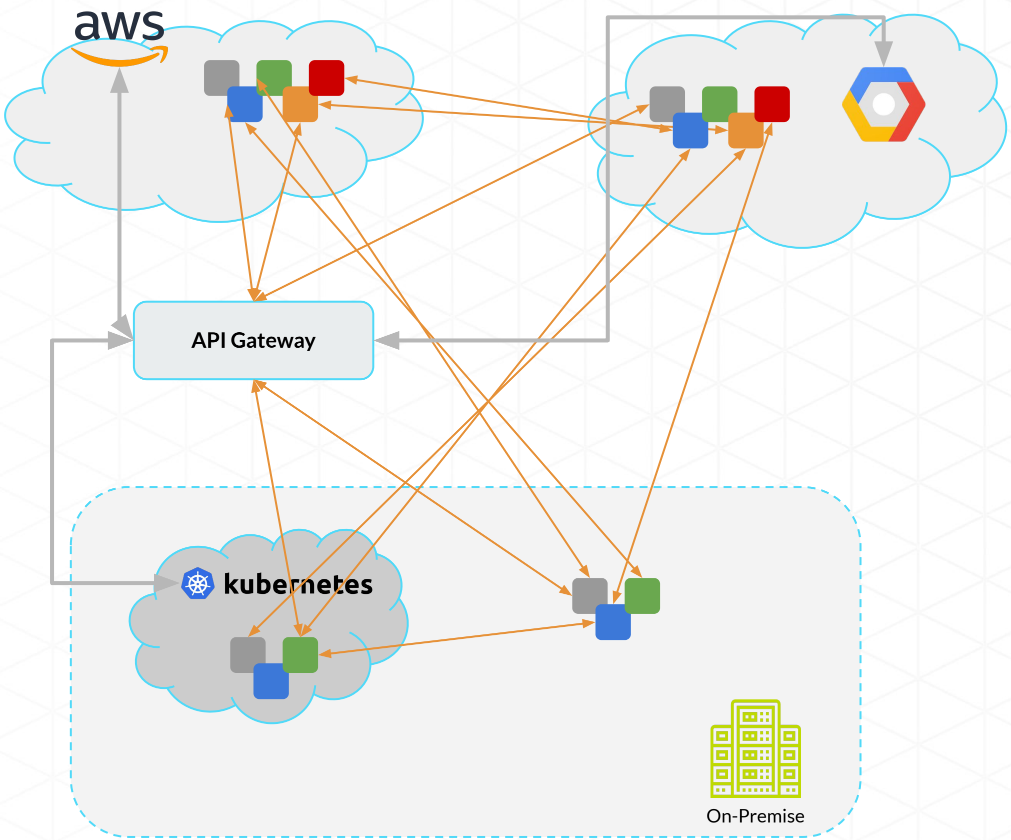
KubeCon NA 2019
San Diego, California





Using SPIRE to connect to third party systems

Let's put **SPIFFE** to practice!

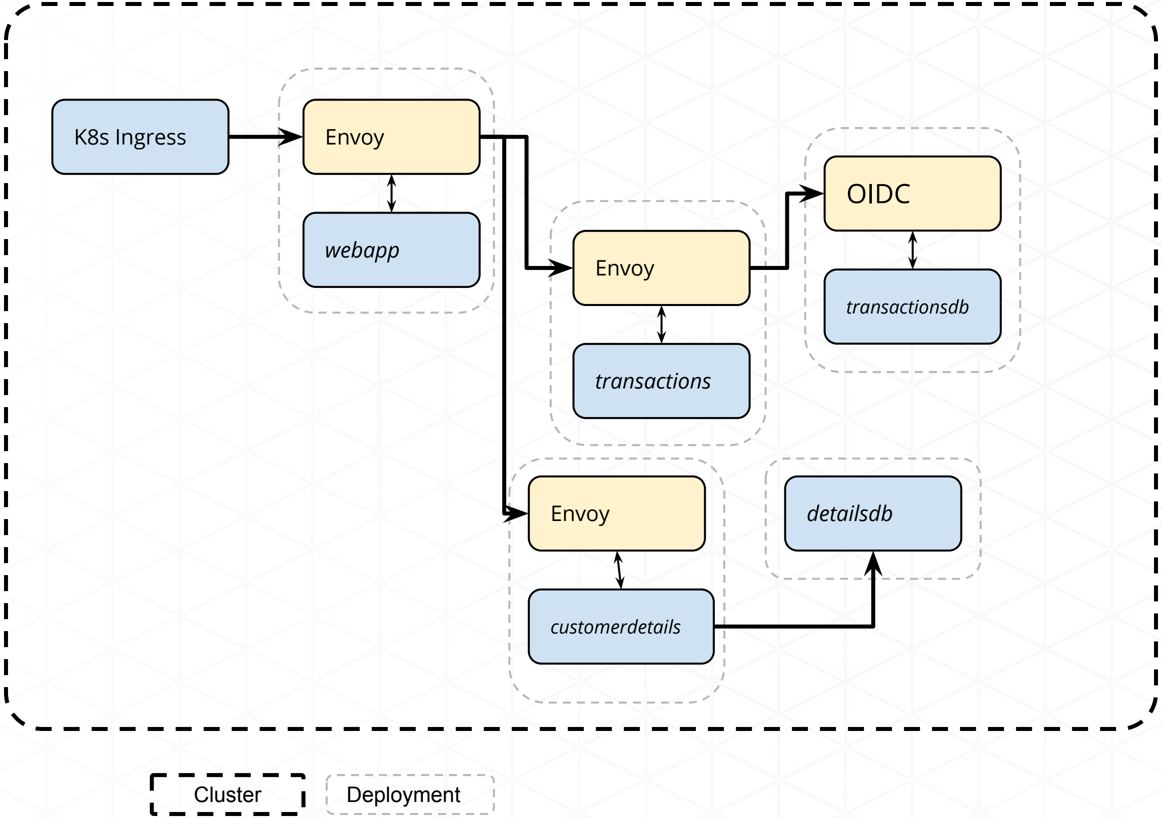


Demo Application

Inter-service using SPIFFE/SPIRE (we are using Envoy to make this really easy)

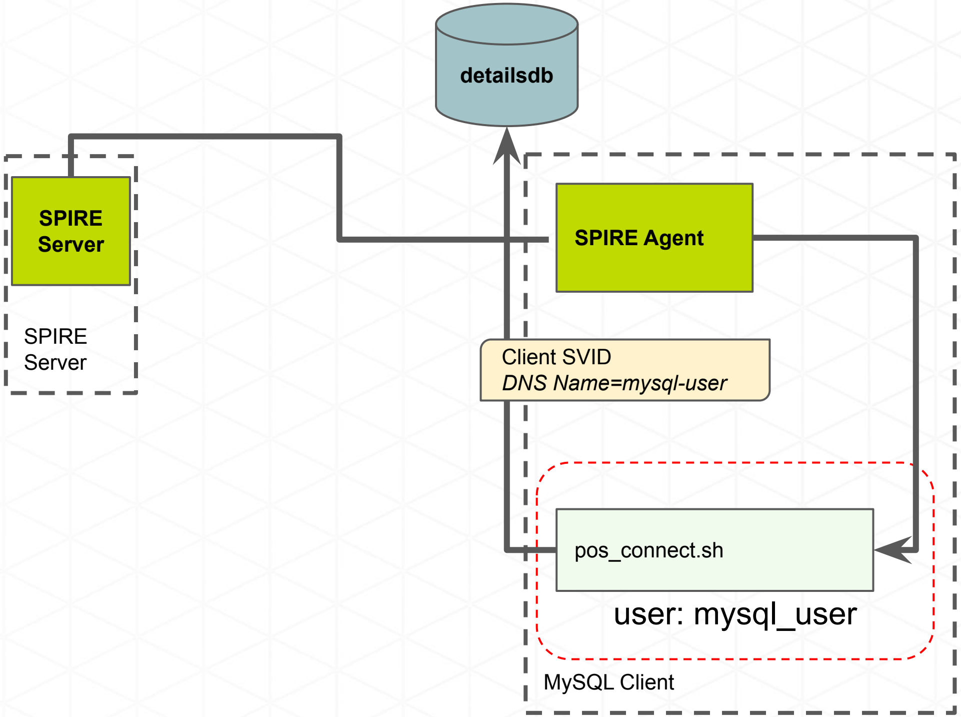
Connecting to postgres (via X.509 authentication)

Connecting cross-cloud to AWS RDS (via OIDC)



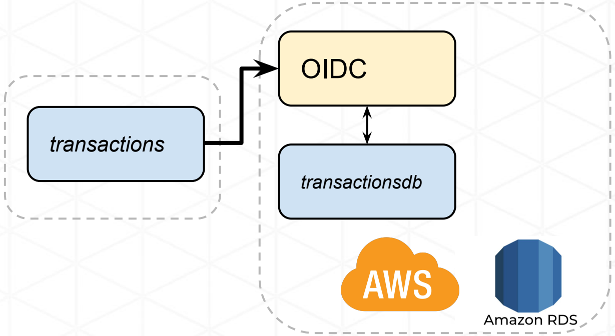
X.509 Authentication to Postgres

MySql authentication is configured to only accept a valid x509 certificate where the certificate's subject name matches the requirement for the MySql account.



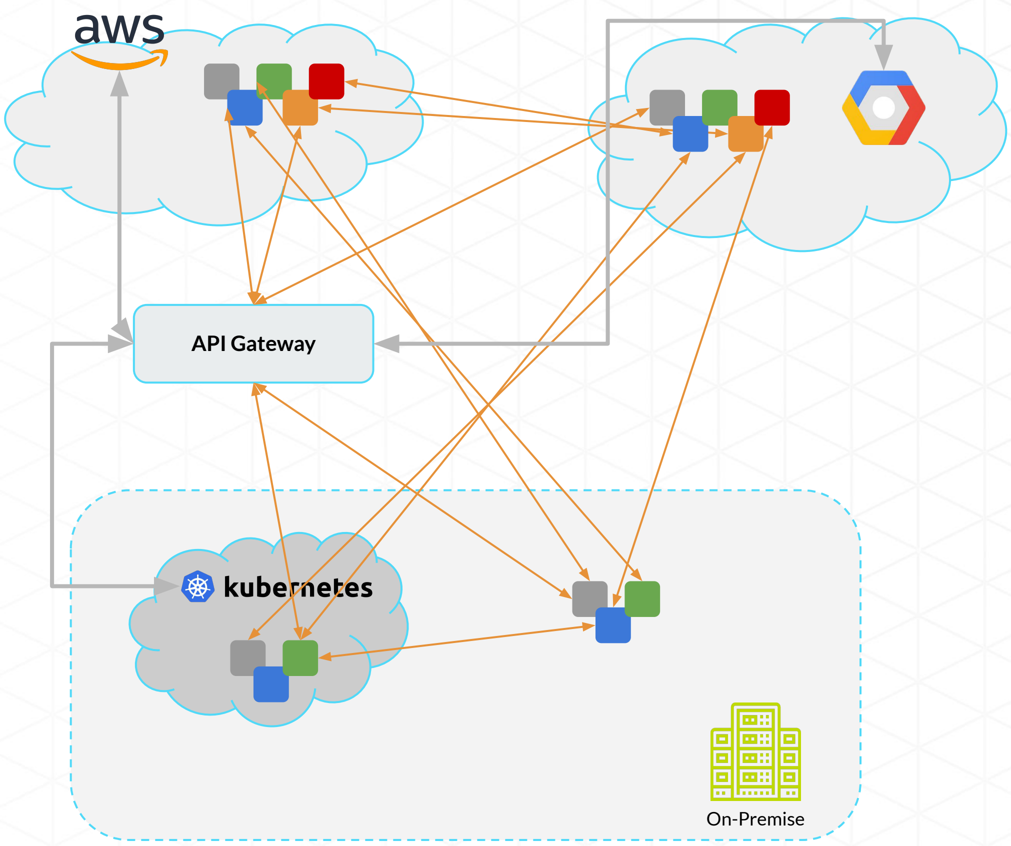
AWS RDS via OIDC

1. SPIRE Server acquires PKI Cert from Let's Encrypt
2. AWS pulls the OpenID discovery document from Scytale Server
3. SPIRE Agent mints a JWT SVID. The AWS SDK sends it to the AWS IAM/STS Assume Role API.
4. The AWS OpenID provider interface fetches the JWKS file from the SPIRE Server.
5. The JWT SVID is verified with the JWKS key. AWS IAM confirms that the requested role is allowed, and mints an STS token for it.
6. The AWS SDK uses the S3 API with the STS token to access the S3 bucket with the assumed IAM role.



We’ve shown you how SPIFFE can connect you to:

- Workloads (ok, maybe you knew that one)
- Between Kubernetes and a VM
- To service providers that support X.509 authentication (there’s plenty!)
- To a cloud provider via OIDC (AWS is supported today, Azure and GCP plan to ship this shortly)
- And we talked about connecting to other service mesh too.



Where next?

spiffe.io

spiffe.slack.com

github.com/spiffe/spire

@ Kubecon

Wednesday 11.50am

Tyler Julian talking about how SPIRE scales at Uber

Thursday 2.25pm

Google Istio team talking about the SPIFFE Federation API

All week

SPIFFE Lightning talks @ the Scytale booth in the sponsor showcase from AWS, Uber, Strya (OPA), ByteDance, Joe Beda and more..