SPIFFE AND SPIRE IN PRACTICE

DANIEL FELDMAN
UMAIR KHAN

AGENDA

1. RECAP: SPIFFE AND SPIRE
2. SECURE MICROSERVICES COMMUNICATION
3. BUILD AND BRIDGE SERVICE MESH
4. AUTHENTICATE SECURELY TO COMMON PLATFORMS
5. AUTHENTICATION FOR ZERO TRUST SECURITY
6. REDUCING THE RISK OF ROGUE CONTAINERS

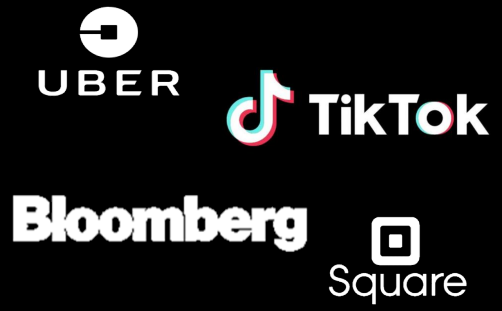
SPIFFE AND SPIRE INTRODUCTION

INTRODUCING SPIFFE AND SPIRE

Open-source specification and toolchain for service identity

Part of CNCF

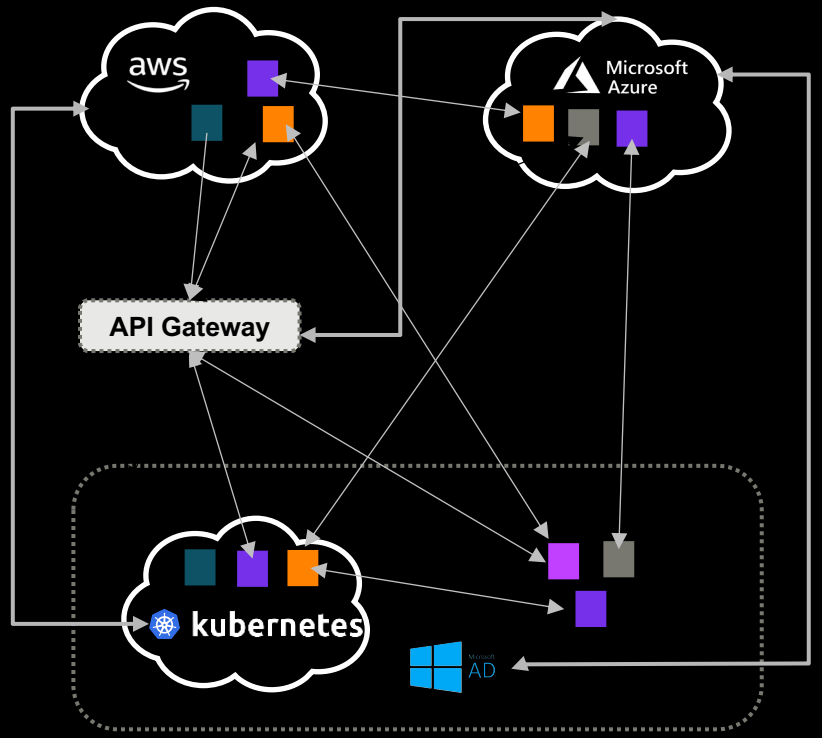
Integrated into various open-source projects

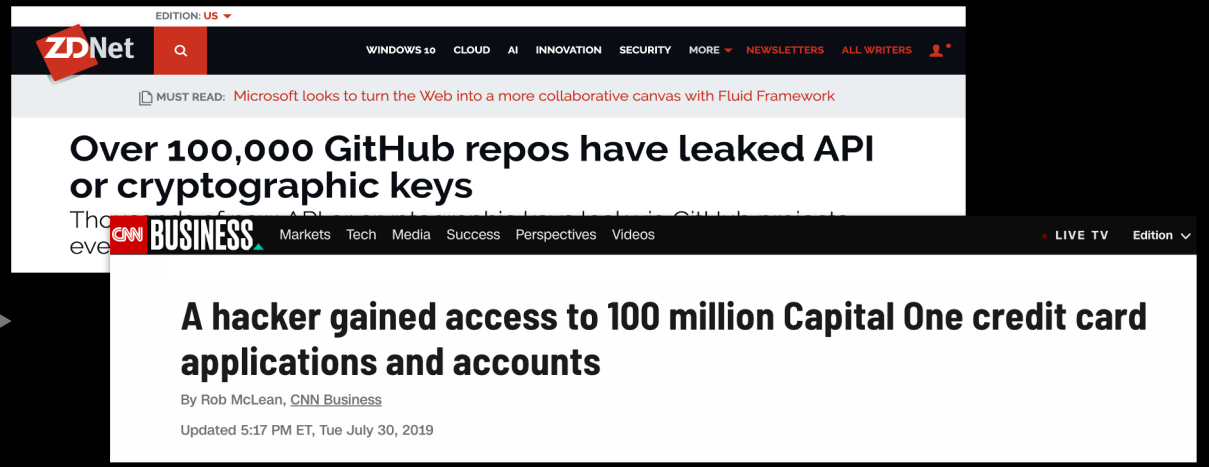


Extensive contributions by HPE and other top tech companies

CROSS-SERVICE COMMUNICATION IS EXPLODING

Increasing attack surface & risk of leakage across untrusted networks

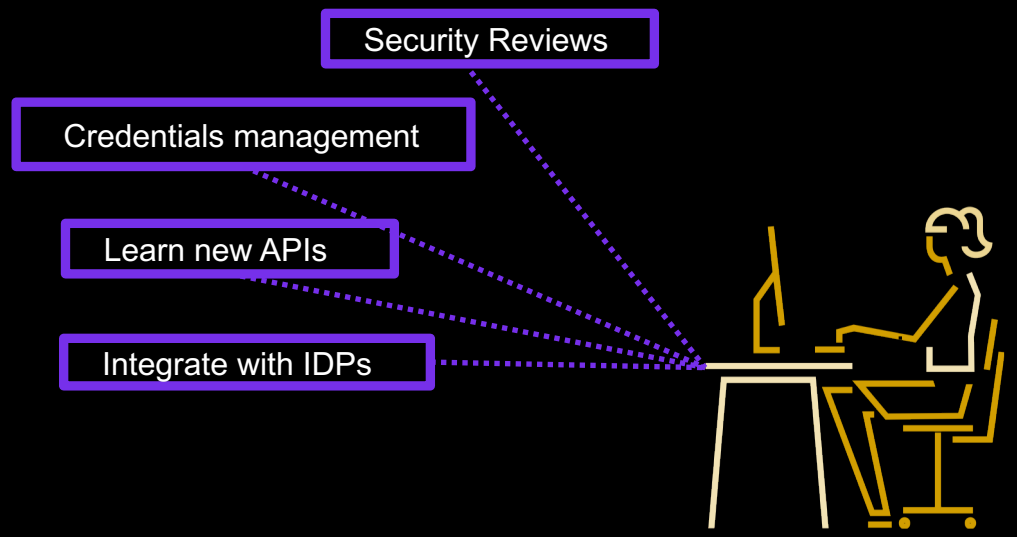




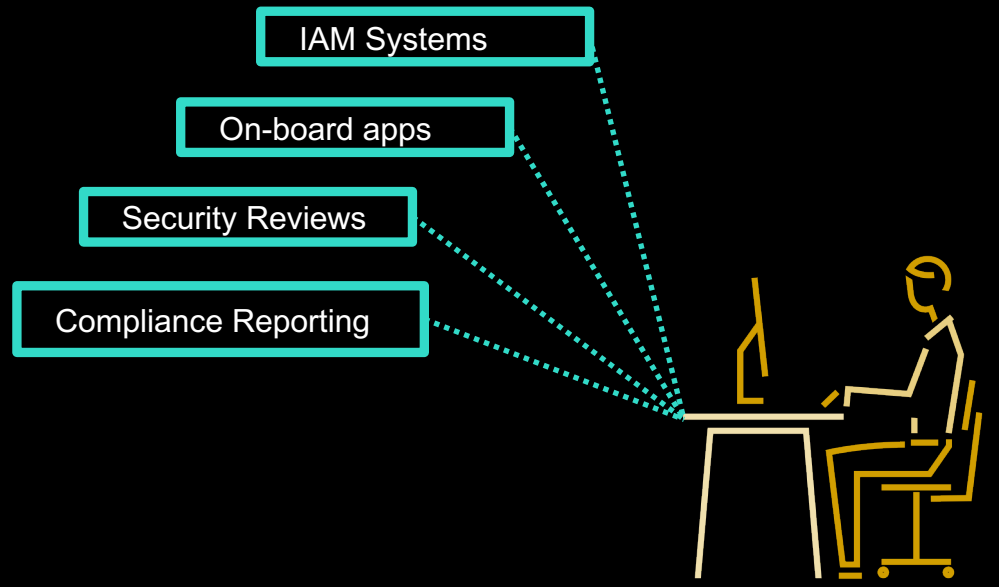
Long-lived service credentials exist across applications, repositories, platforms, and tools, making them ripe for theft.

CROSS-SERVICE COMMUNICATION IS EXPLODING

Increasing operational complexity and reducing developer velocity



Developers of Apps / Platforms



Security Operation

SOLVING THE “BOTTOM TURTLE”

Service Certificate

Platform Secret Store

Platform Identity

Initial Configuration?

SOURCE: SOLVING THE BOTTOM TURTLE: WWW.SPIFFE.IO/BOOK

SOLVING THE “BOTTOM TURTLE”

Service Certificate

Platform Secret Store

Platform Identity

Initial Configuration?

SOURCE: SOLVING THE BOTTOM TURTLE: WWW.SPIFFE.IO/BOOK

SPIFFE KEY CONCEPTS

SPIFFE ID

Standard format for a service identifier

spiffe://trustdomain/service

SPIFFE VERIFIABLE IDENTITY DOCUMENT

Cryptographically verifiable document asserting a SPIFFE ID

TRUST BUNDLE

Set of public keys used to verify SVIDs

WORKLOAD API

Local API for workloads to retrieve their SPIFFE IDs, SVIDs, and Trust Bundles

SPIRE

Core Differentiators

MULTI-FACTOR ATTESTATION

Has it been signed by the CI/CD pipeline?

Is it known to trusted middleware or schedulers?

Is the machine a member of a known network or cluster?

Can we affirm the integrity of the machine it runs upon?

- Real time, attestation engine issues and validates cryptographic service identifies (SPIFFE) based on multiple factor policy
- Eliminates the need for secret management

AUTOMATED LIFECYCLE MANAGEMENT

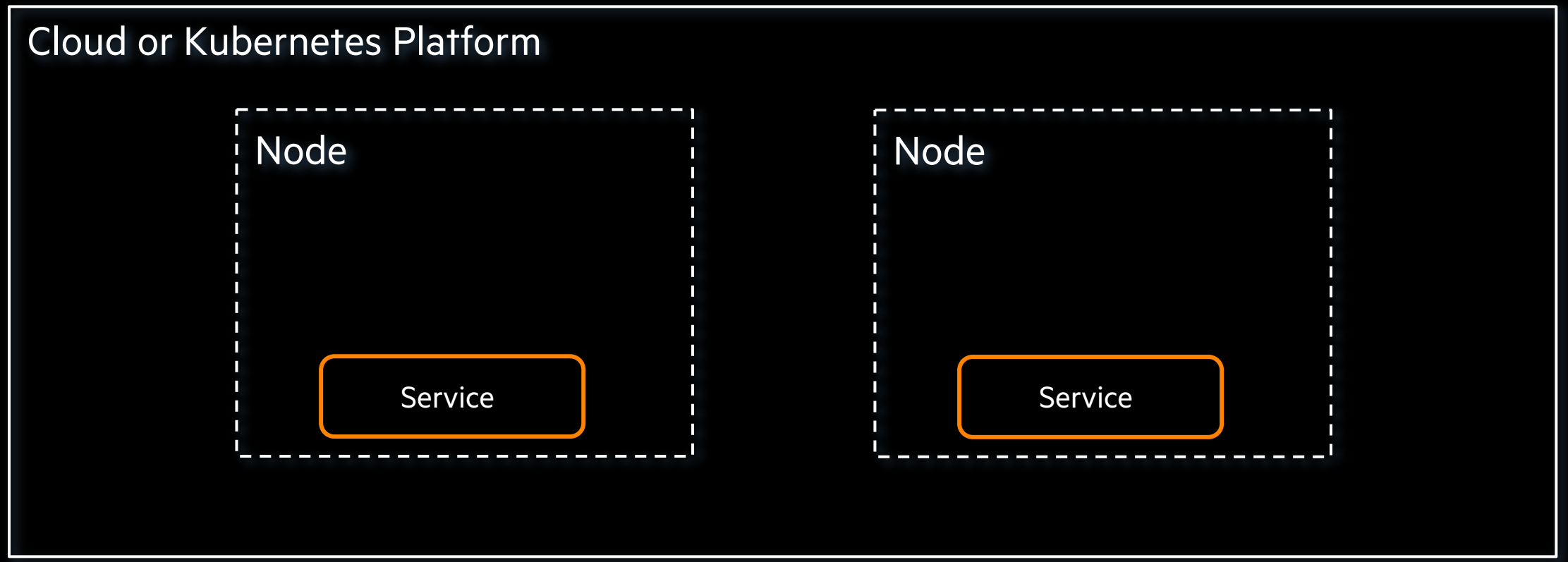
- Automatically issues, distributes, and renews short-live credentials
- Reduces operational overhead associated with credential management

EXTENSIBLE, WEB-SCALE ARCHITECTURE

- Easily extends to identity providers, certificate authorities, and systems
- Designed for dynamic, distributed environments

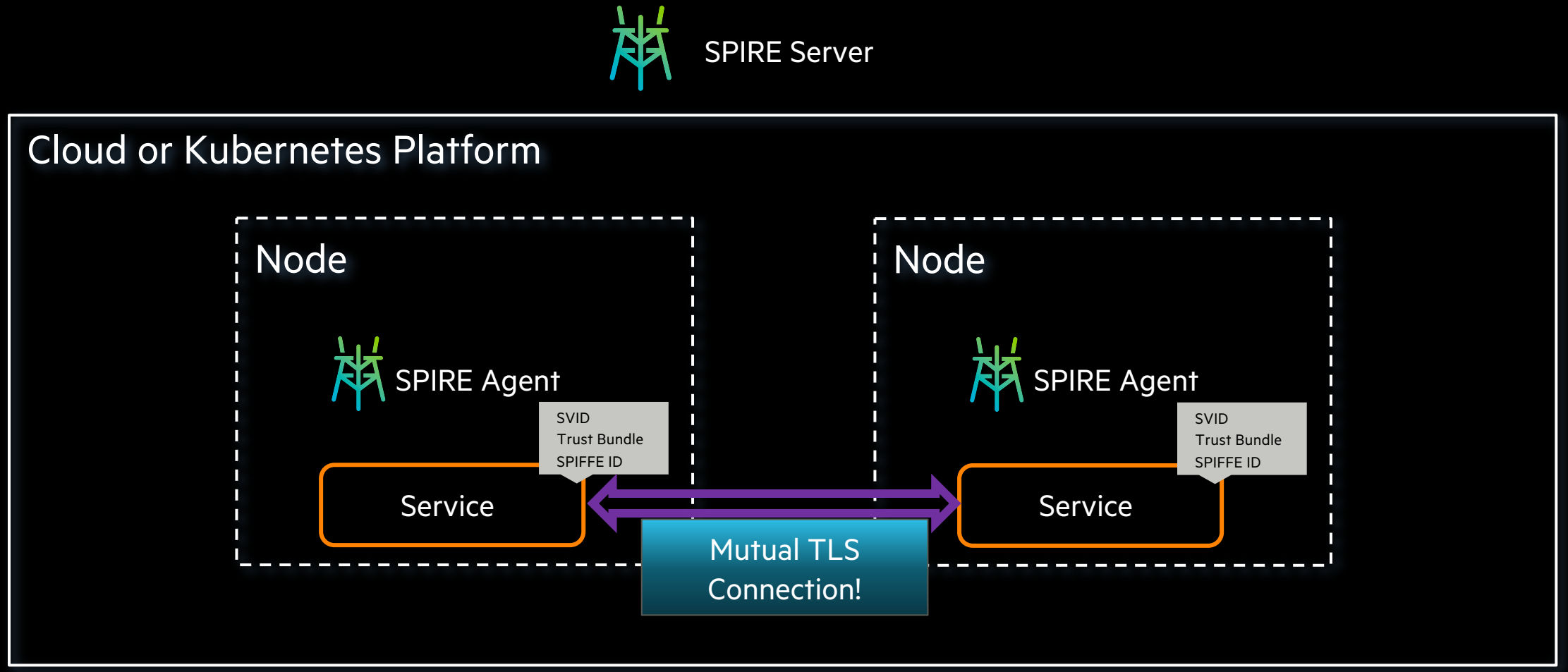
SECURE MICROSERVICES COMMUNICATION

SECURE MICROSERVICES COMMUNICATION

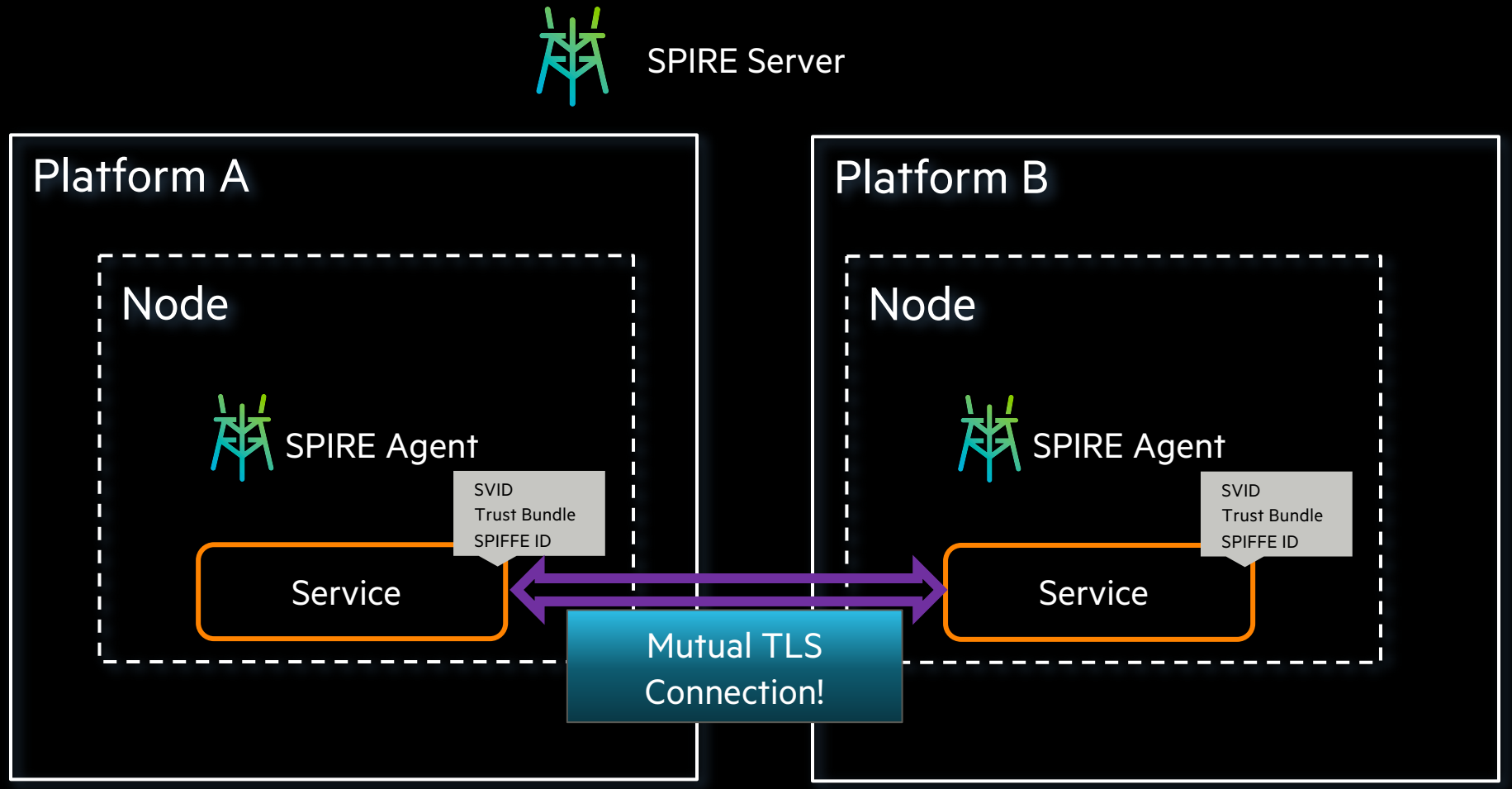


SECURE MICROSERVICES COMMUNICATION

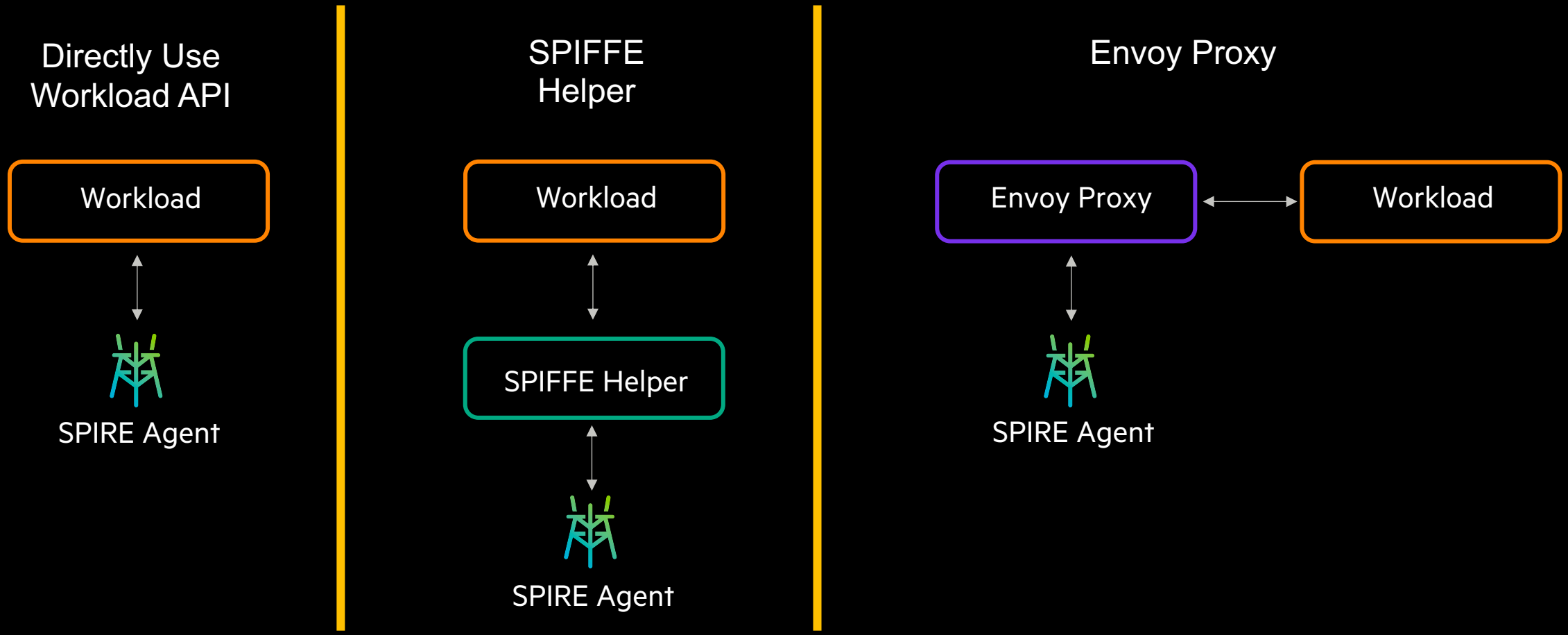
SECURE MICROSERVICES COMMUNICATION



SECURE MICROSERVICES COMMUNICATION



IMPLEMENTATION OPTIONS



SERVICE MESH COMPARISON

- Works across different service meshes and outside service meshes
- Can do hardware-level or cloud-level attestation
- More fine-grained control over certificates

UBER: SECURING NEXT-GEN AND LEGACY INFRASTRUCTURE

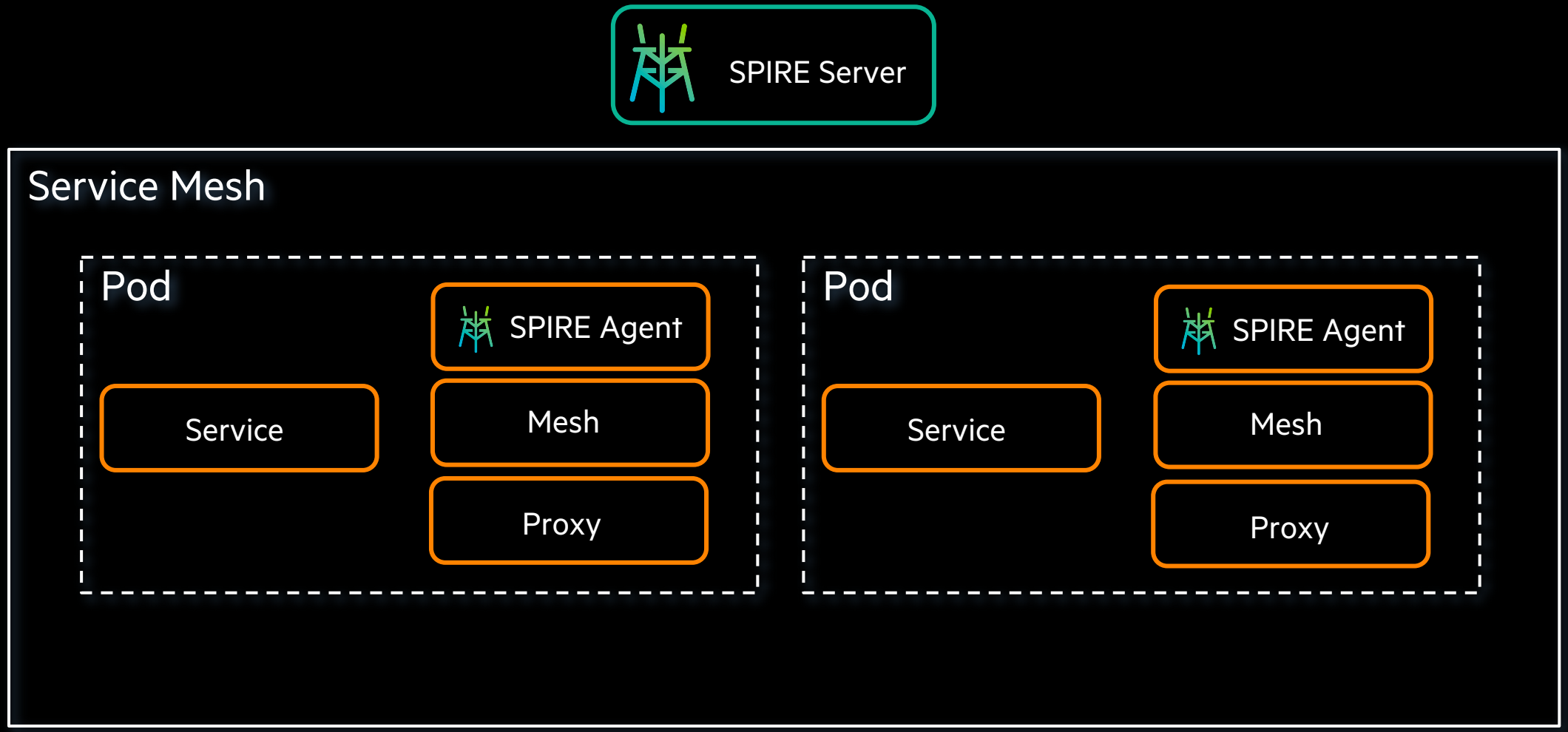
"SPIRE is now a key component of Uber's next infrastructure, but we are also using a side-car approach to retrofit authentication into legacy infrastructure. While SPIFFE and SPIRE are commonly known to work in modern, cloud native architectures, we can adapt the projects to our proprietary legacy stack quickly. SPIRE can provide a critical bridge of trust within Uber's next-gen and legacy infrastructure and positively impact internal security and developer efficiency"

Ryan Turner, Software Engineer 2, Uber

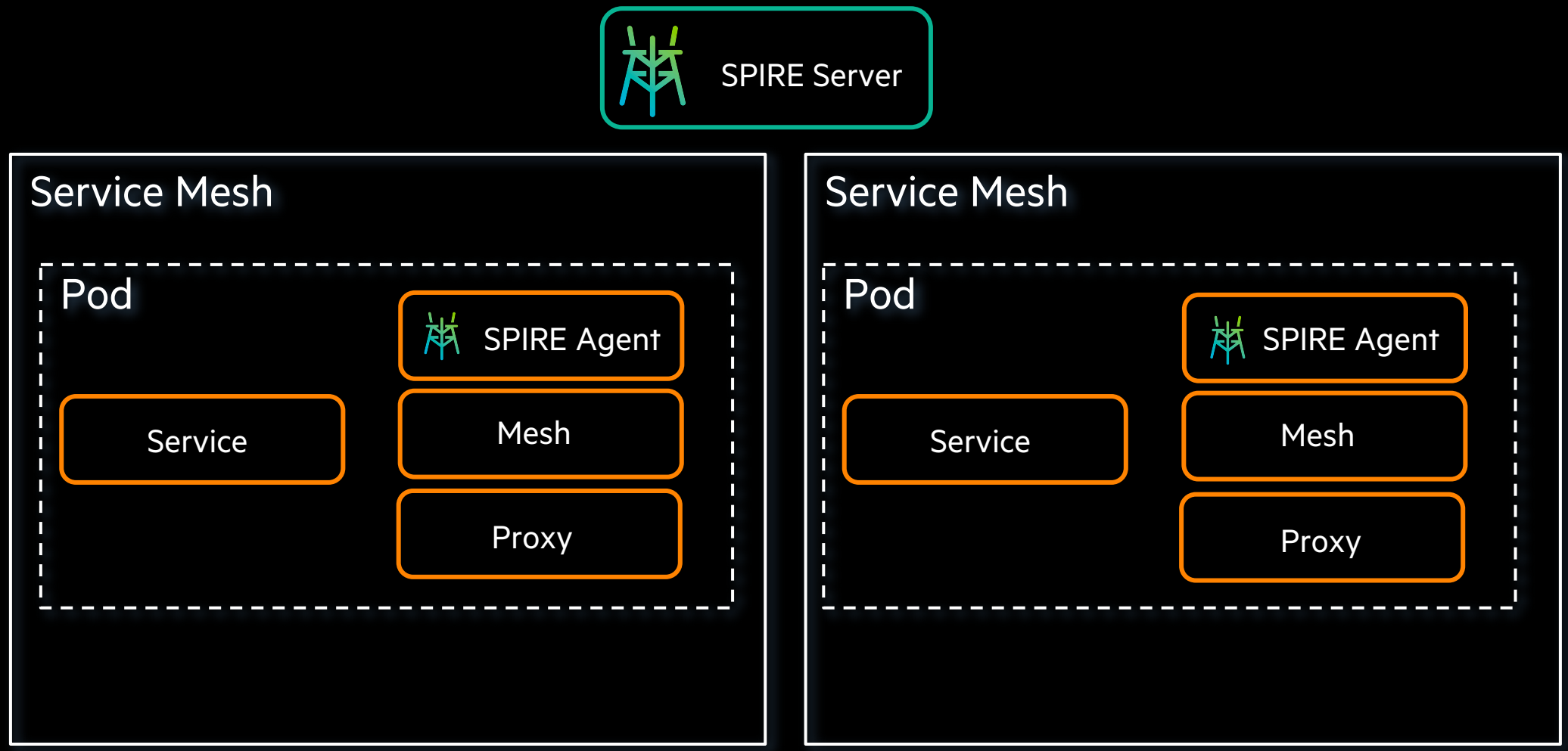
Source: Solving the bottom turtle: www.spiffe.io/book

BUILD AND BRIDGE SERVICE MESH

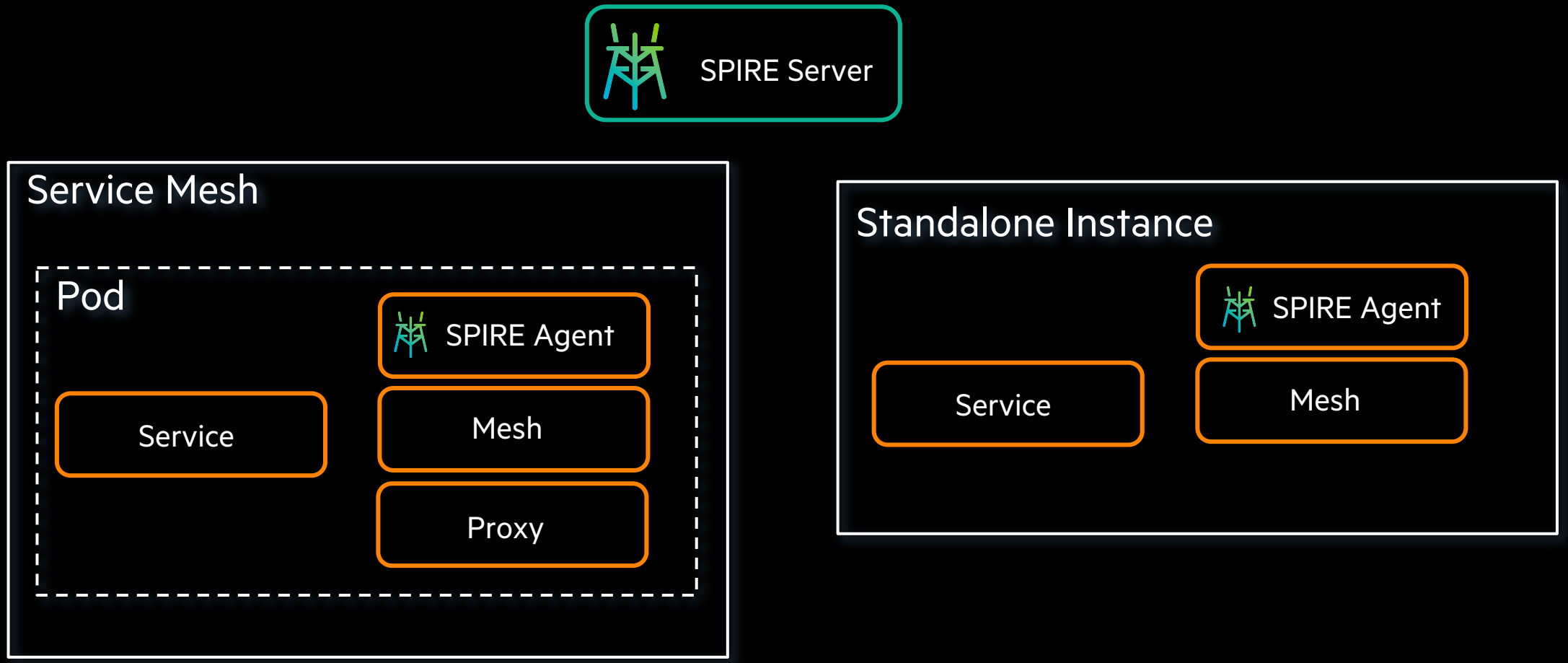
BUILD AND BRIDGE SERVICE MESH



BUILD AND BRIDGE SERVICE MESH



BUILD AND BRIDGE SERVICE MESH



SOME EXAMPLES

- AWS App Mesh
- GreyMatter
- Istio
- Kuma
- Network Service Mesh
- NGINX Service Mesh
- Open Service Mesh

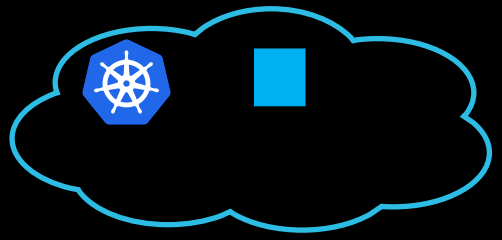
AUTHENTICATE SECURELY TO DATABASES OR CLOUD PLATFORM

AUTHENTICATE TO COMMON DATABASES OR CLOUD PLATFORMS

Reduce reliance on passwords or API keys

Using usernames/passwords or tokens to access resources outside Kubernetes e.g. Datastores

- Risk of breach is higher
- Need to generate/manage credentials

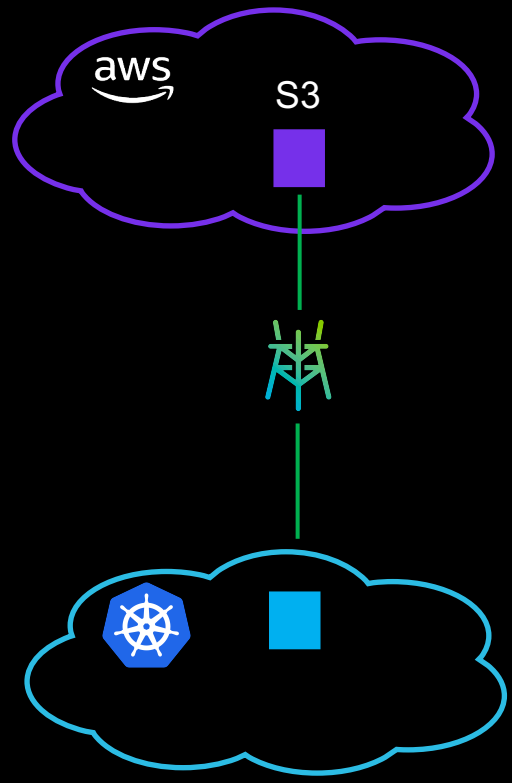


AUTHENTICATE TO COMMON DATABASES OR CLOUD PLATFORMS

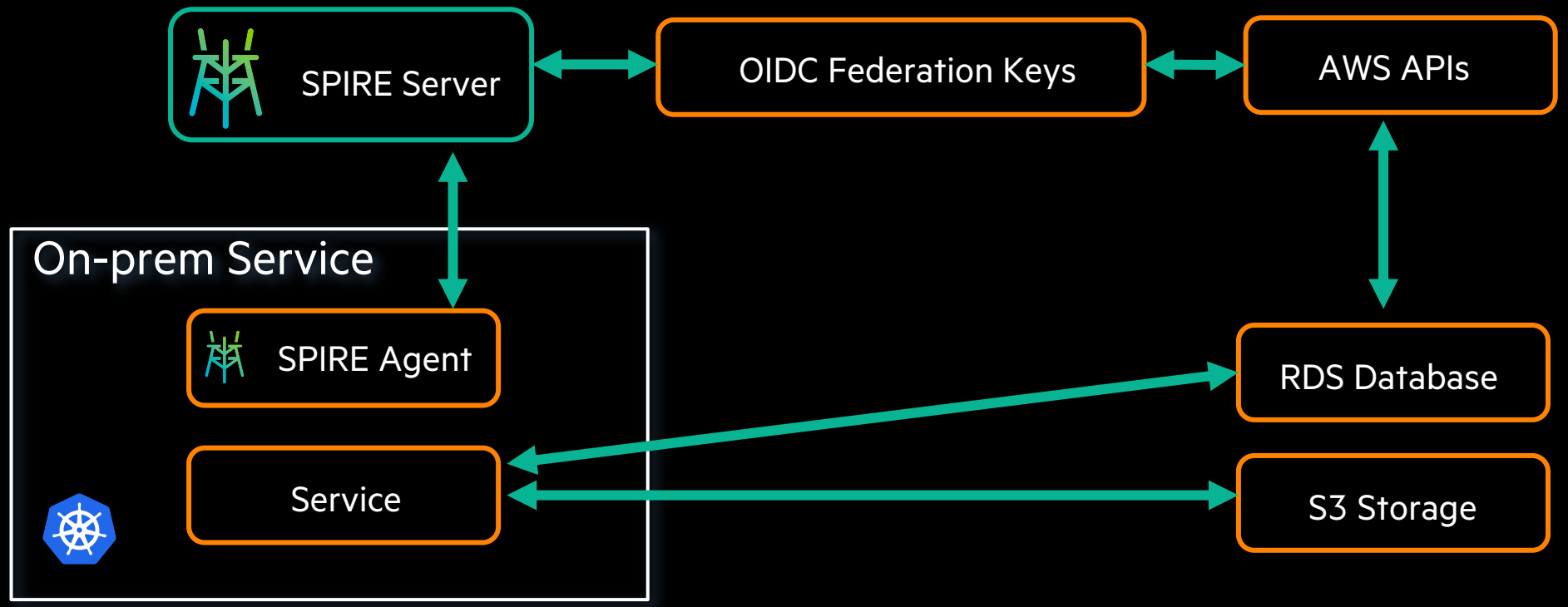
Reduce reliance on passwords or API keys

With SPIRE you can:

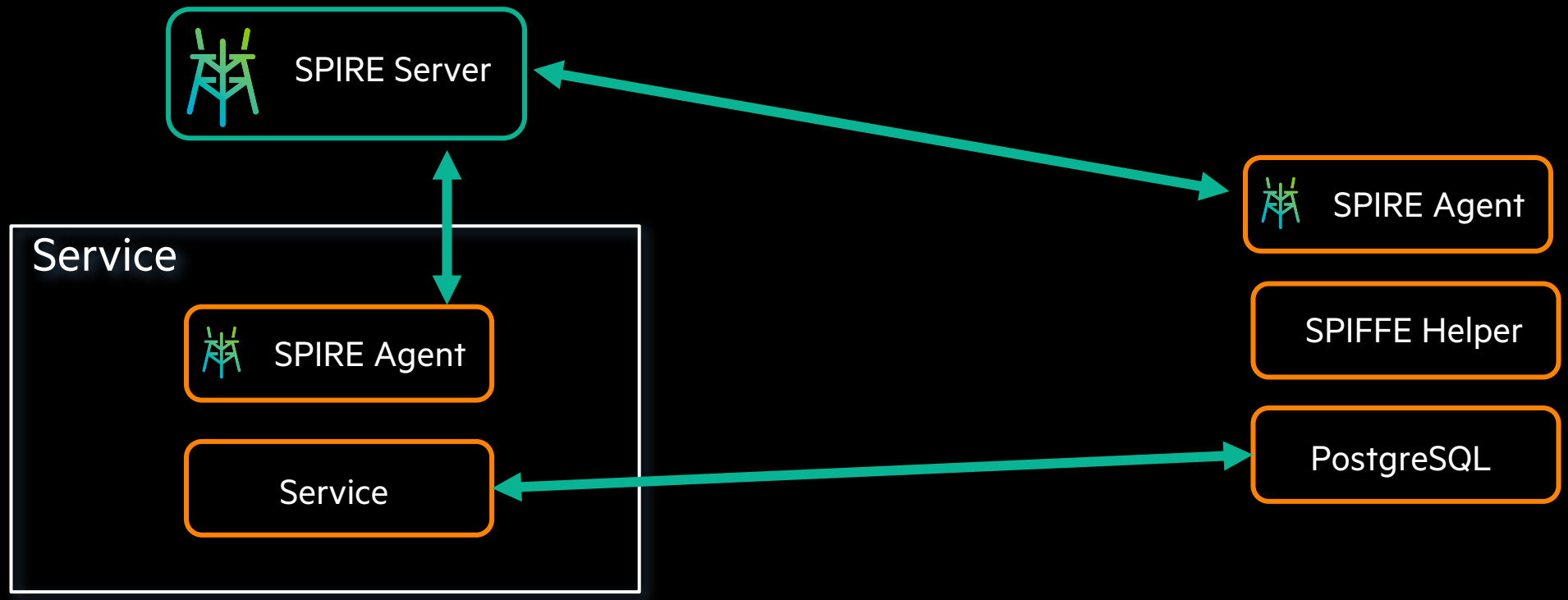
- Eliminate the need to generate and manage distinct shared secrets for each cloud platform for each application.
- Scale, secure identity-driven authentication across all cloud providers and platforms.



AUTHENTICATE TO AWS



AUTHENTICATE TO POSTGRESQL



Spiffe-helper pushes certificates into PostgreSQL
Services can authenticate as PostgreSQL users

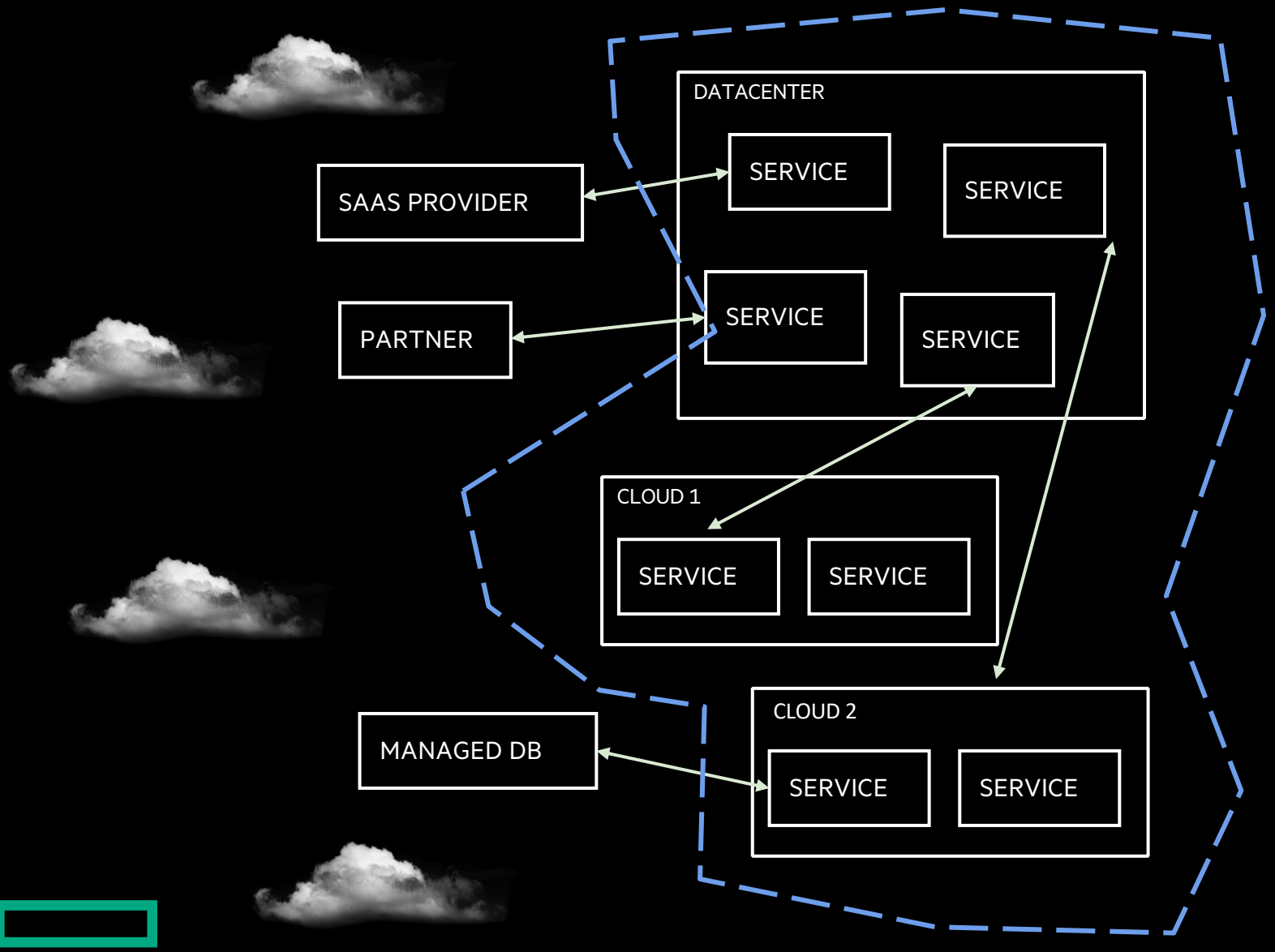
SCROOGE MCBANK: CUSTOMER PORTAL

Your current balance is 9.65

Description	Debit	Credit
Daily parking	$20	
Gliderport Paragliding	$165	
San Diego Zoo Daily pass	$65	
Airline Compensation		$300
Carne Asada Street Tacos	$24.65	
Coin-Op Game Room	$35	

AUTHENTICATION FOR ZERO TRUST SECURITY MODEL

PERIMETER SECURITY



As we add:

- services
- datacenters
- clouds
- regions inside clouds

perimeter security becomes increasingly untenable

CLOUD AND CONTAINERS DRIVING ADOPTION OF ZERO TRUST

Traditional network based security models don't work in modern software architectures

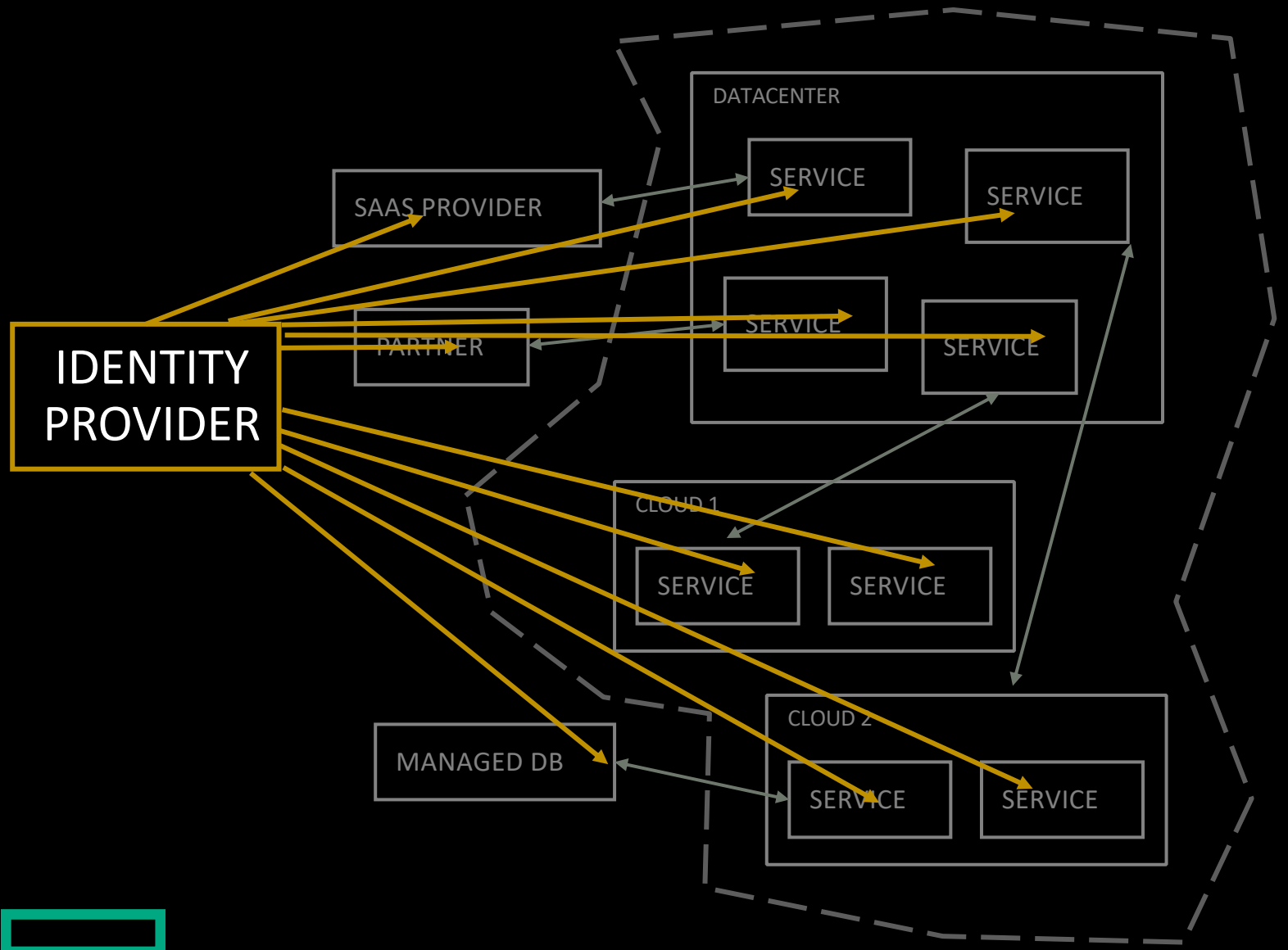
Perimeter based

- Attempts to build a trusted "wall"
- Relies on IP addresses or physical locations
- Difficult to implement for today's dynamic environments

Zero Trust

- Assumes "bad guys" are everywhere
- Uses cryptographic identities for authenticating every system/user
- Enables universal enforcement across hybrid infrastructures

SPIFFE IS FOUNDATIONAL FOR ZERO TRUST SECURITY MODEL



Each service gets its own
- unique
- secure
- provable

identity

ANTHEM: BUILDING A FOUNDATION FOR ZERO TRUST IN HEALTHCARE

"We **could not rely on traditional parameter-based** security tools and processes to secure our next-generation applications and infrastructure**. Zero trust,** a fine-grained, automated approach to security, made a lot of sense to us, especially in the future, as we plan to **operate across organizational boundaries and cloud providers**. Identity and authentication for both users and services are among the zero trust security model's core principles. Zero trust allows us to rely less on network-based controls than authenticating every system or workload**. SPIFFE and SPIRE have enabled a foundational authentication layer** for our zero trust security architecture. They allow each workload to cryptographically prove "who they are" before they start communicating"

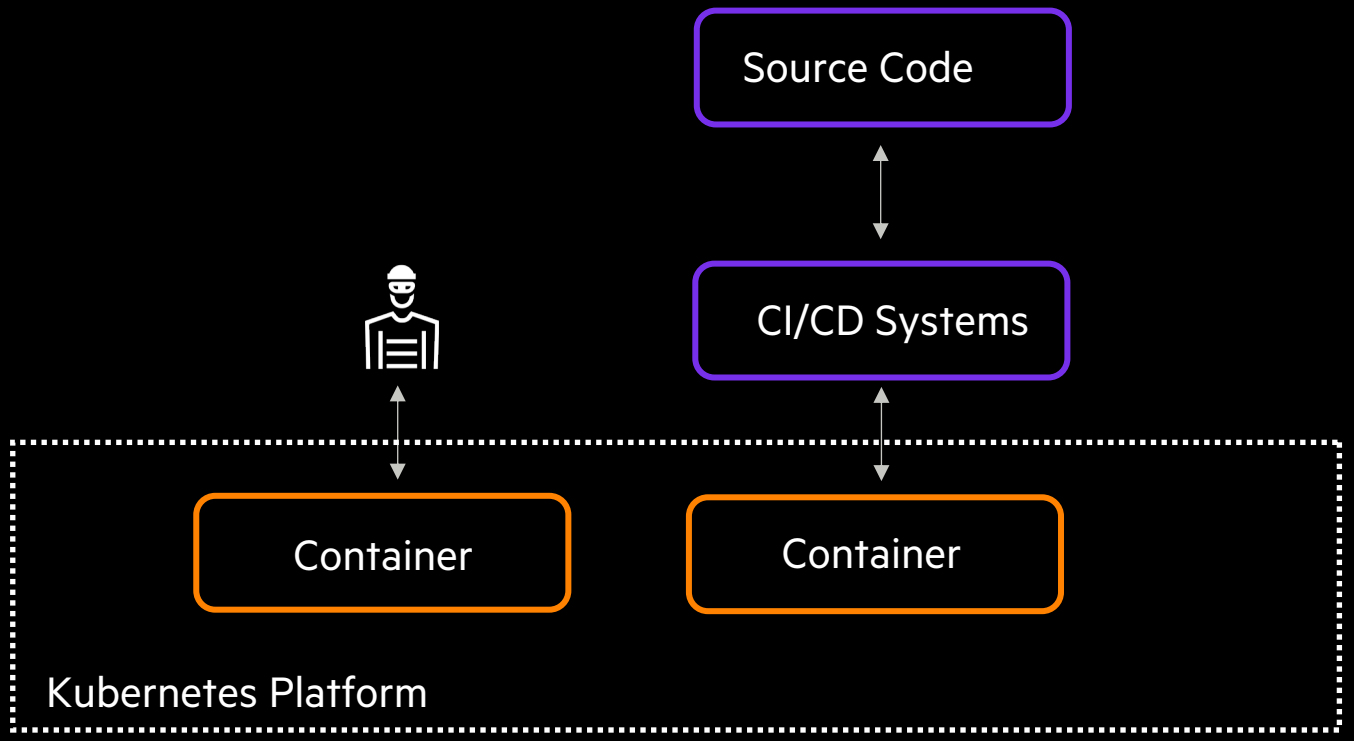
Bobby Samuel, VP AI Engineering, Anthem

Source: Solving the bottom turtle: www.spiffe.io/book

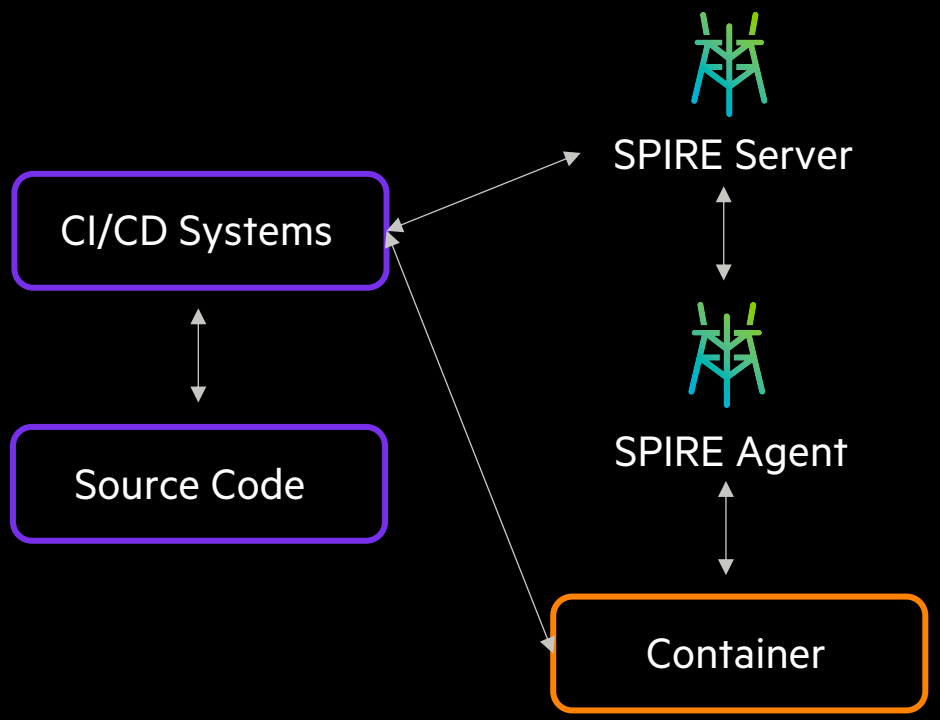
REDUCING THE RISK OF ROGUE CONTAINERS

CHALLENGE : REDUCE RISK OF ROGUE CONTAINERS

Attackers can insert new containers or workloads

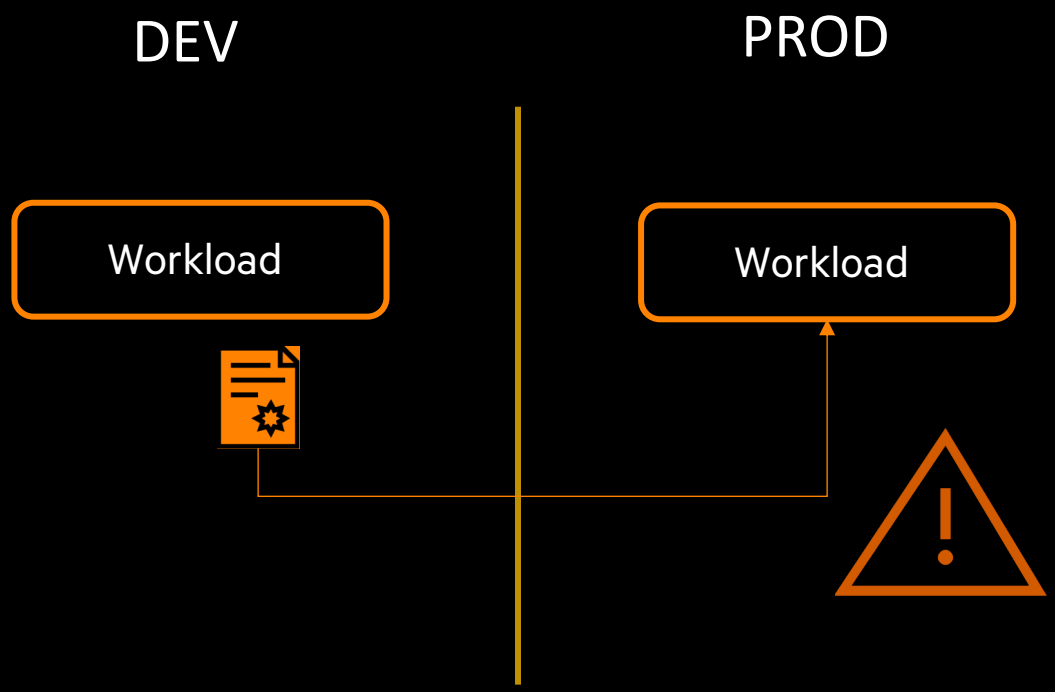


SOLUTION: BINARY ATTESTATION WITH SPIRE

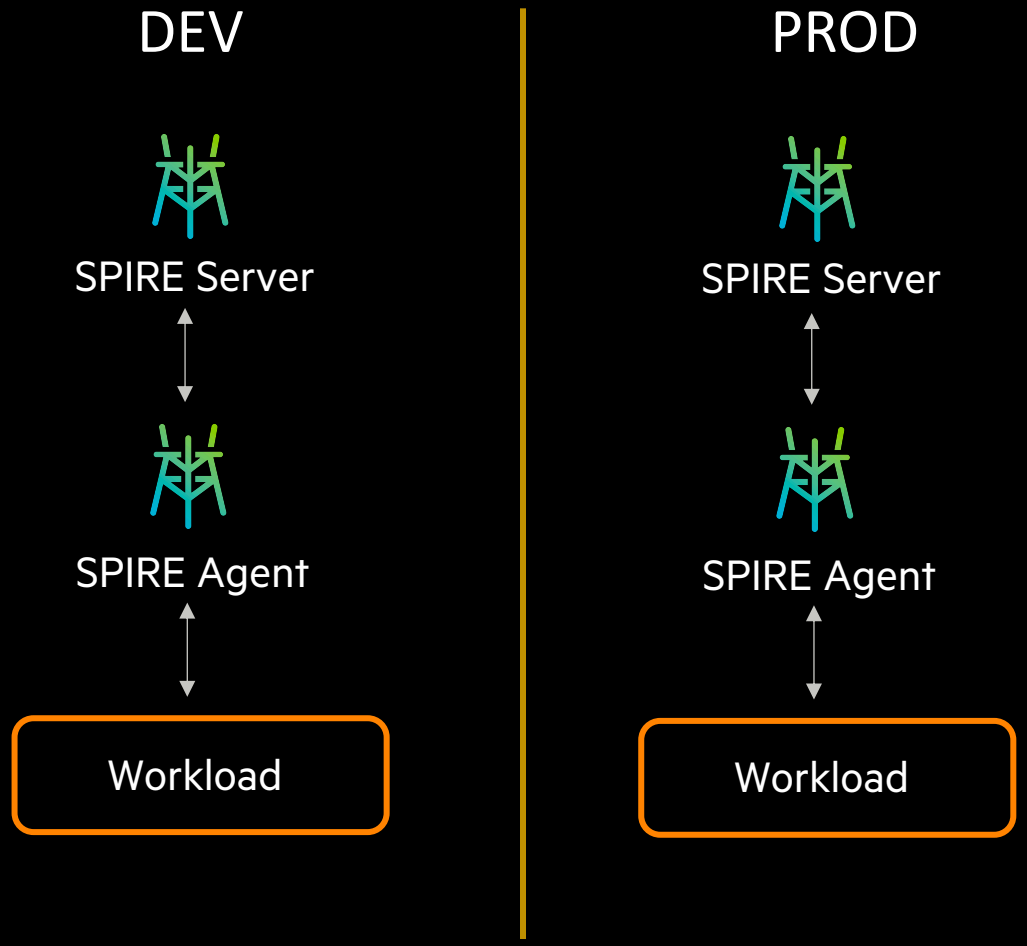


- Each time the CICD system builds a container image, it sends the container hash to SPIRE
- SPIRE then checks the container hash each time it grants an identity document
- This guarantees attackers can't insert new containers, modify containers in the container registry, or bypass CI/CD security checks

CHALLENGE : SERVICE DISRUPTION DUE TO MISCONFIGURATION



SOLUTION: REDUCE RISK OF MISCONFIGURATION



SEPARATE TRUST DOMAINS

Use separate SPIRE domains for dev and prod workloads, in order to ensure isolation of prod workloads.

TRANSPARENT AUTHENTICATION HAS SIMPLIFIED OPERATIONS

"With SPIRE we can deploy a consistent, "dial-tone" authentication across all our platforms. The burden of authentication and security is now encapsulated from the developers so they can focus on business or application logic. This has improved our deployment velocity overall. We are also less likely to get "production errors" due to configuration issues such as using development credentials in production. Standardized authentication with SPIRE has also simplified compliance and audit since we have mutual TLS across trust domains and platforms.

Eli Nesterov, Security Engineering Manager, ByteDance

Source: Solving the bottom turtle: www.spiffe.io/book

THANK YOU

Slack.spiffe.io

SPIFFE.io

spiffe.io/book

github.com/spiffe