Save Your Services From Sneaky Snoops With SPIFFE

Daniel Feldman
SPIFFE & SPIRE Projects

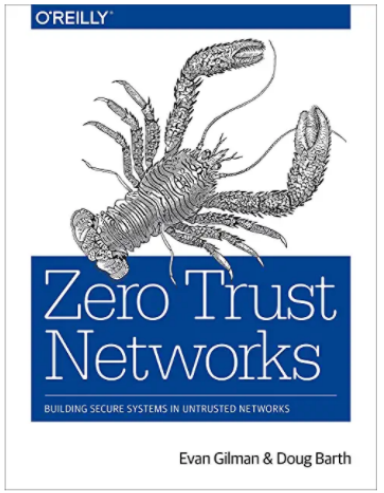
About Me

Daniel Feldman
Cloud security engineering
Veritas -> Scytale -> HPE

And a huge team with security experience from a variety of companies!

aws Google okta

pagerduty DUO



Why Service Identity

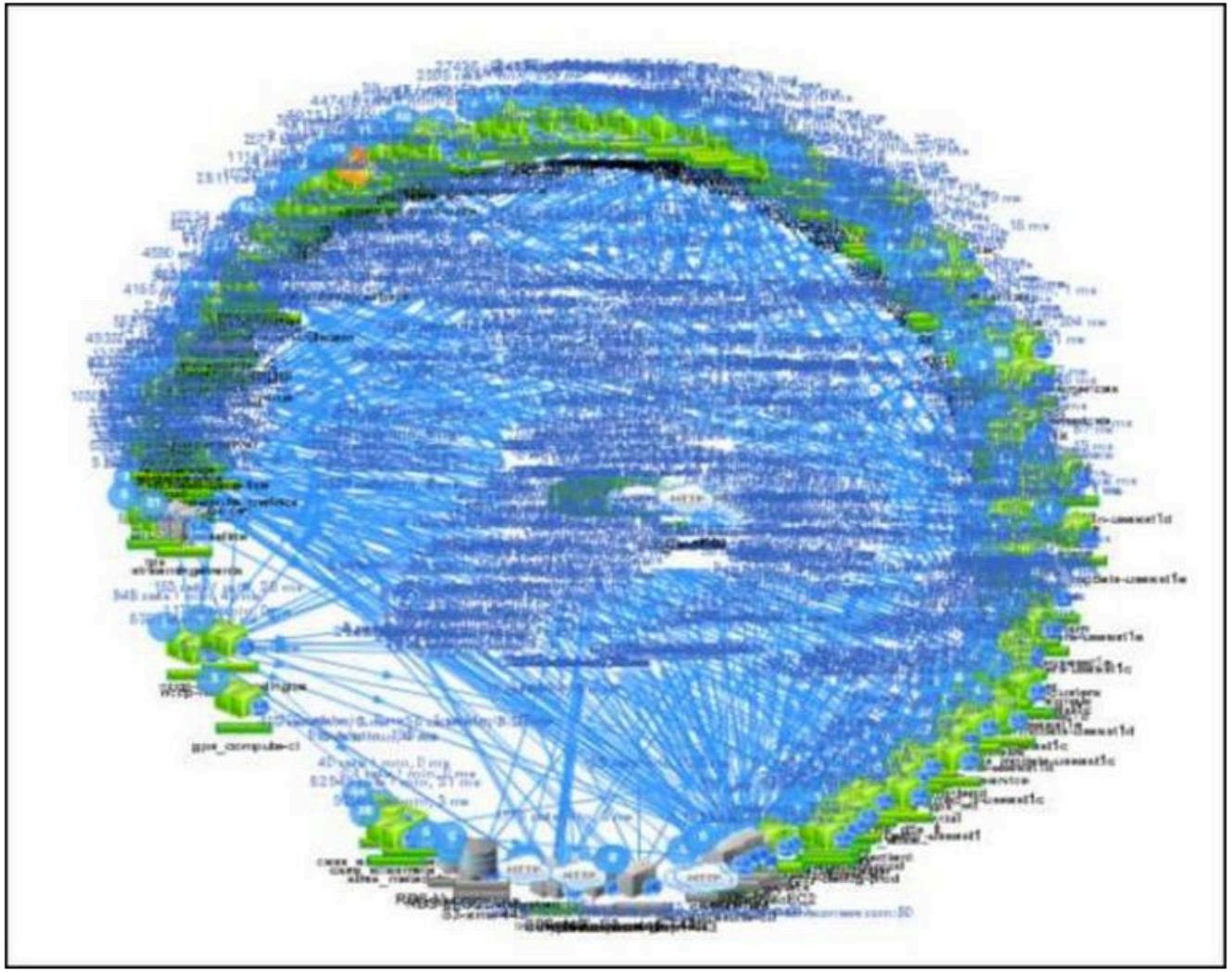
Service Identity Approaches

SPIFFE (the standard)

SPIRE (the implementation)

Next Steps

Why Service Identity

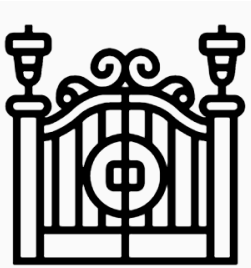


Why Service Identity

ENCRYPTION

INTEGRITY

AUTHENTICATION



AUTHORIZATION

All security rests on a foundation of **identity**

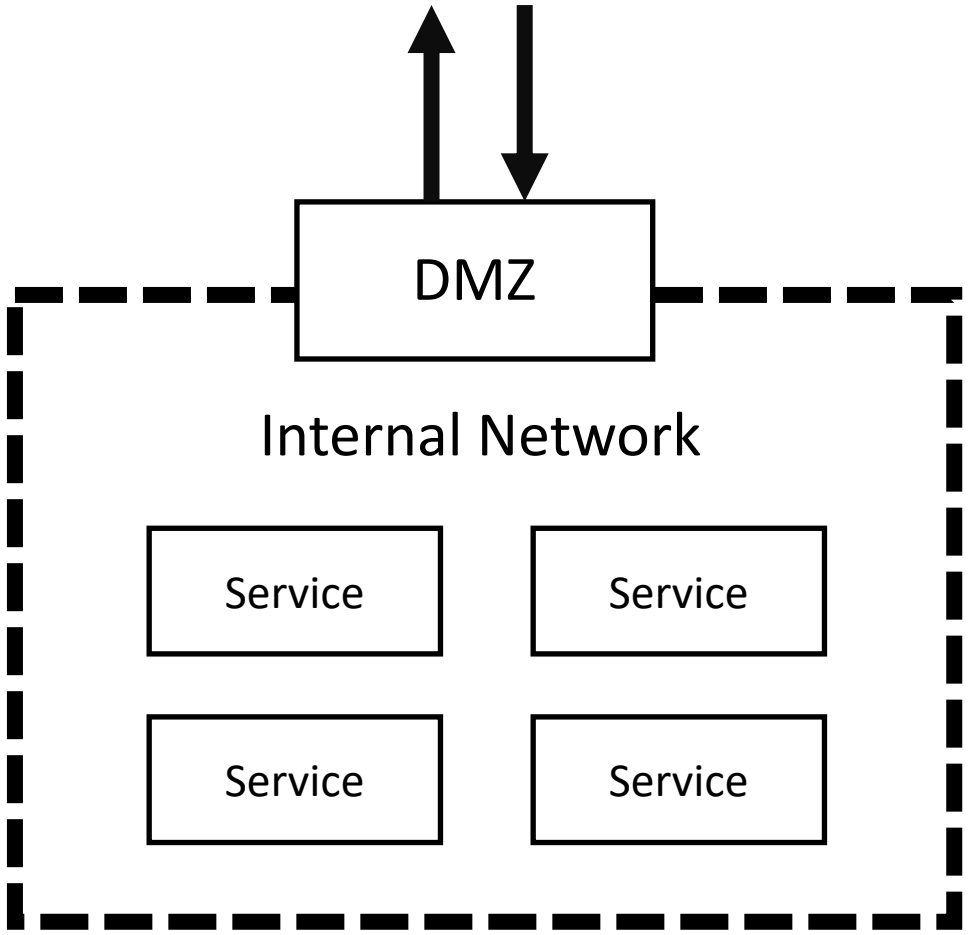
For humans, identity is (mostly) straightforward

For services…

Service identity approaches

KubeCon CloudNativeCon Europe 2020 Virtual

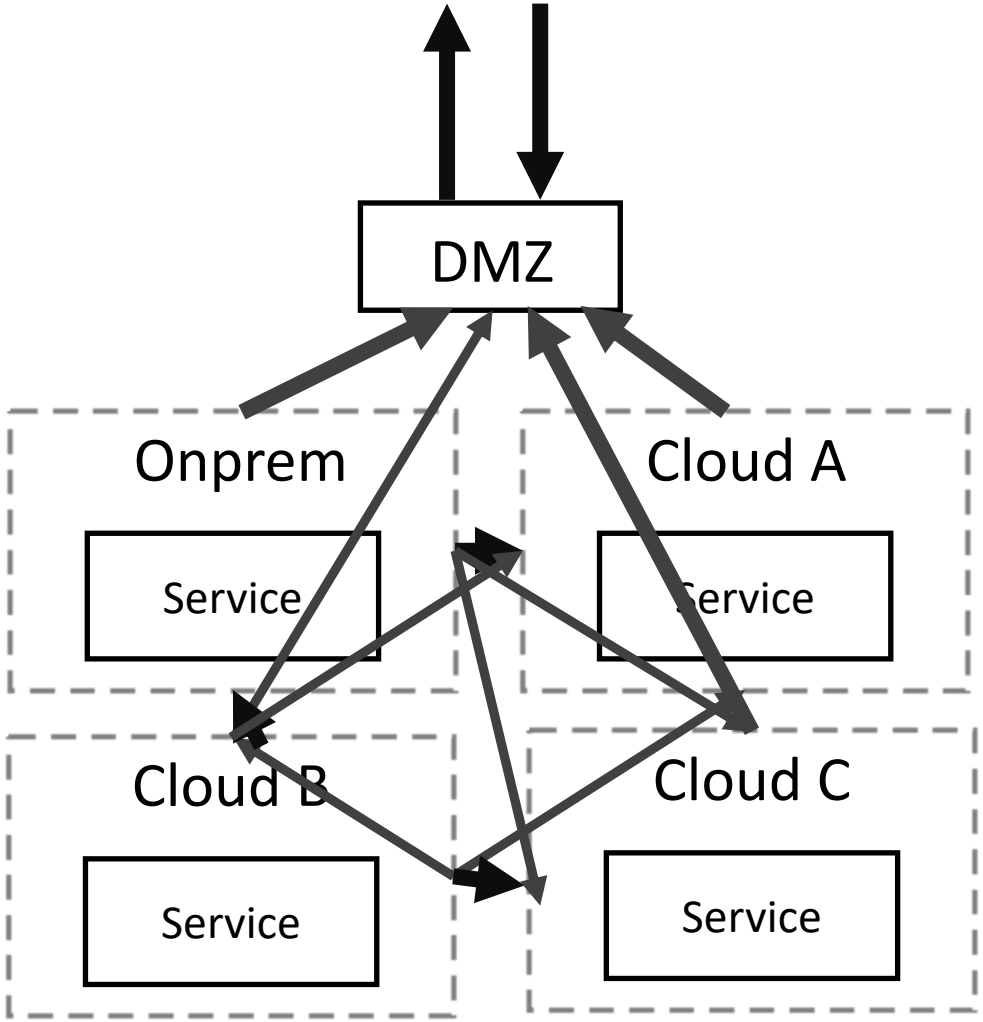
Can we just use… **perimeter security**?



- In the perimeter security model, **everything "inside" is trusted and everything "outside" is untrusted**

Why Service Identity

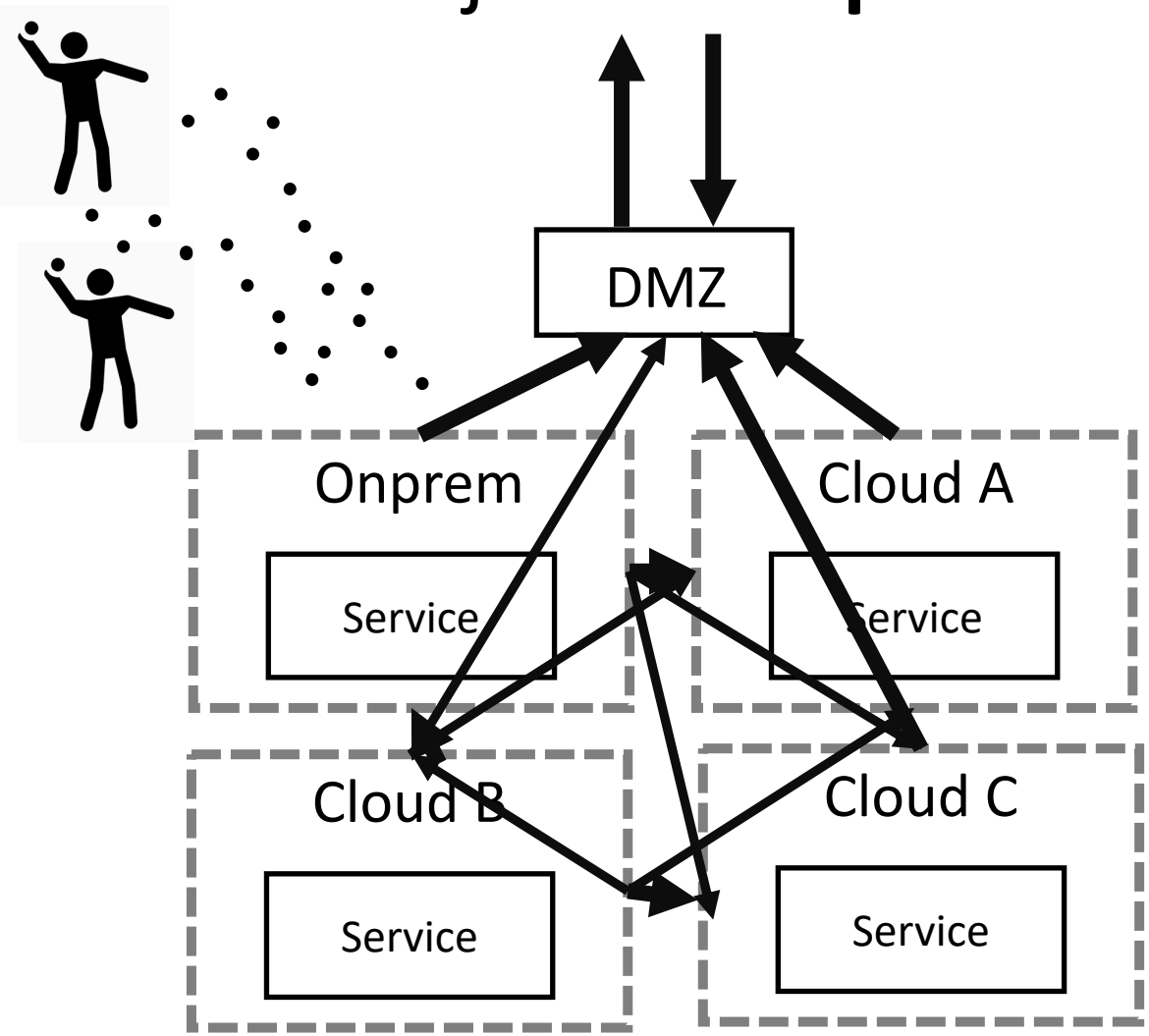
Can we just use… **perimeter security**?



- In the perimeter security model, everything “inside” is trusted and everything “outside” is untrusted
- As you add multiple datacenters, clouds and regions this becomes **increasingly complex**

Why Service Identity

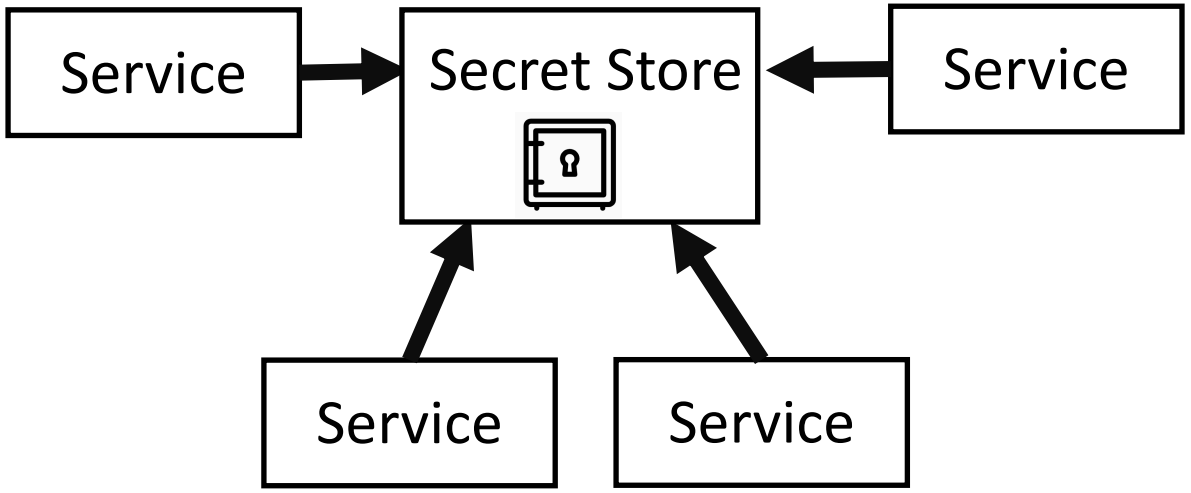
Can we just use… **perimeter security**?



- In the perimeter security model, everything “inside” is trusted and everything “outside” is untrusted
- As you add multiple datacenters, clouds and regions this becomes increasingly complex
- CICD makes it almost impossible

Why Service Identity

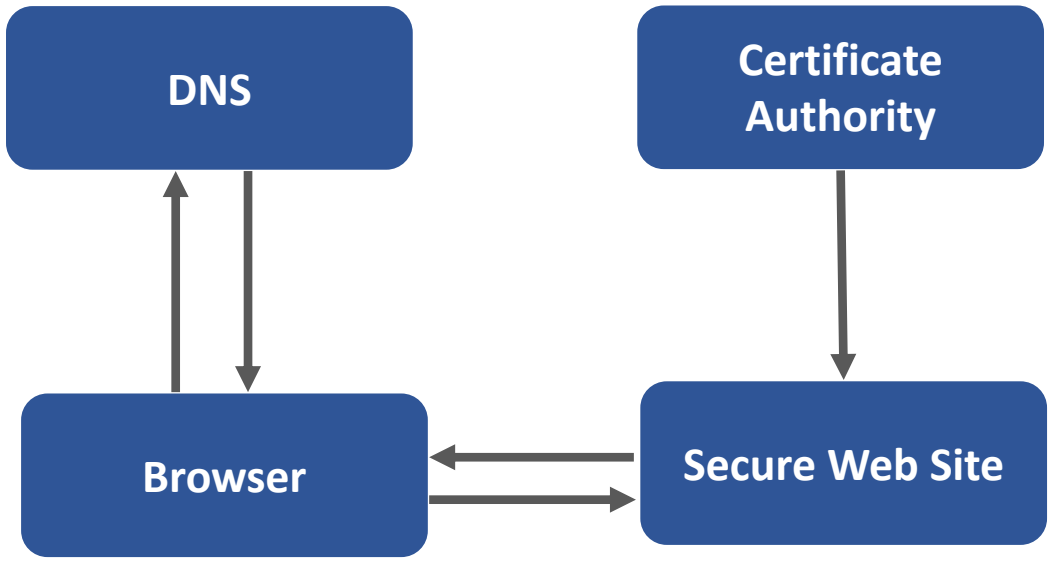
Can we just use… **shared secrets**?



- In the **shared secrets** model, every service has credentials/certificates used to access other services
  - (Hopefully) stored in a secret store
- Maintaining secrets at scale is hard work
- How do you initially authenticate to the secret store?

Why Service Identity

Can we just use… **web PKI**?



- Web Public key infrastructure (PKI) lets you access web sites securely
- But services aren't web sites:
  - Services don't all have hostnames
- Also, no simple way to check the client's identity

Why Service Identity

We need a way to **securely distribute identities to services** without relying on manual effort

SPIFFE (the standard)

Secure Production Identity Framework for Everyone

- SPIFFE is a **platform-agnostic standard** for implementing service identity
- Inspired by Google, Netflix, and Facebook
- It gives your service a “passport” for communicating with other services
- It is partially implemented by several service meshes, and fully implemented in SPIRE

SPIFFE (the standard)

SPIFFE IDs

Standard format for a service ID
spiffe://trustdomain/service

SPIFFE (the standard)

SPIFFE IDs

Standard format for a service ID
spiffe://trustdomain/service

SVIDs

Cryptographically verifiable documents asserting a SPIFFE ID

SPIFFE (the standard)

SPIFFE IDs

Standard format for a service ID
spiffe://trustdomain/service

SVIDs

Cryptographically verifiable documents asserting a SPIFFE ID

Trust Bundles

Set of certificates used to verify SVIDs

SPIFFE (the standard)

SPIFFE IDs

Standard format for a service ID
spiffe://trustdomain/service

SVIDs

Cryptographically verifiable documents asserting a SPIFFE ID

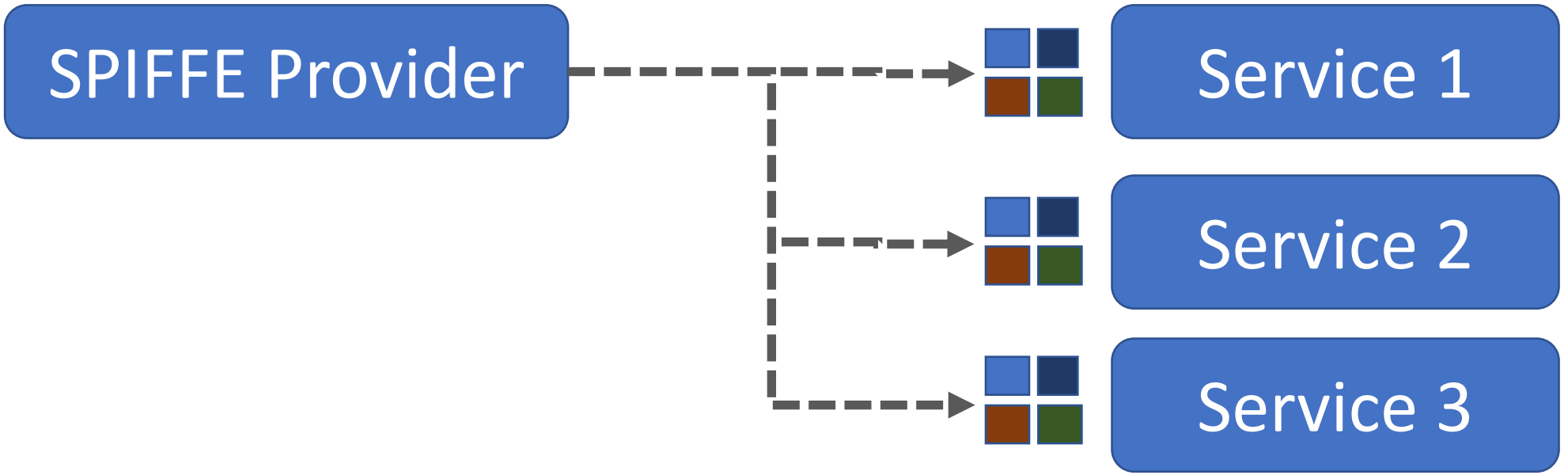
Trust Bundles

Set of certificates used to verify SVIDs

Workload API

Local gRPC API for workloads to obtain SPIFFE IDs, SVIDs, and Trust Bundles

SPIFFE (the standard)

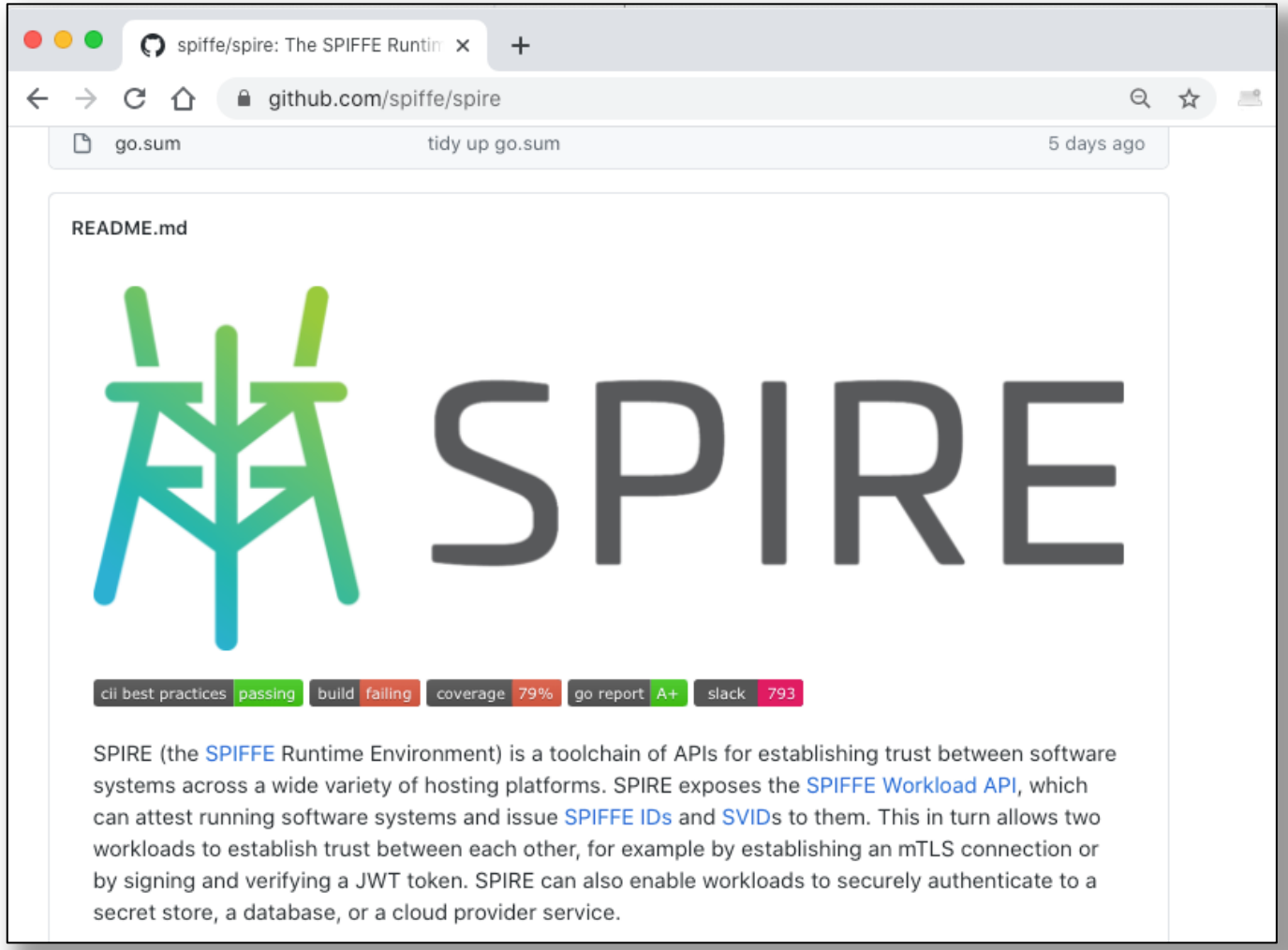


Kuma

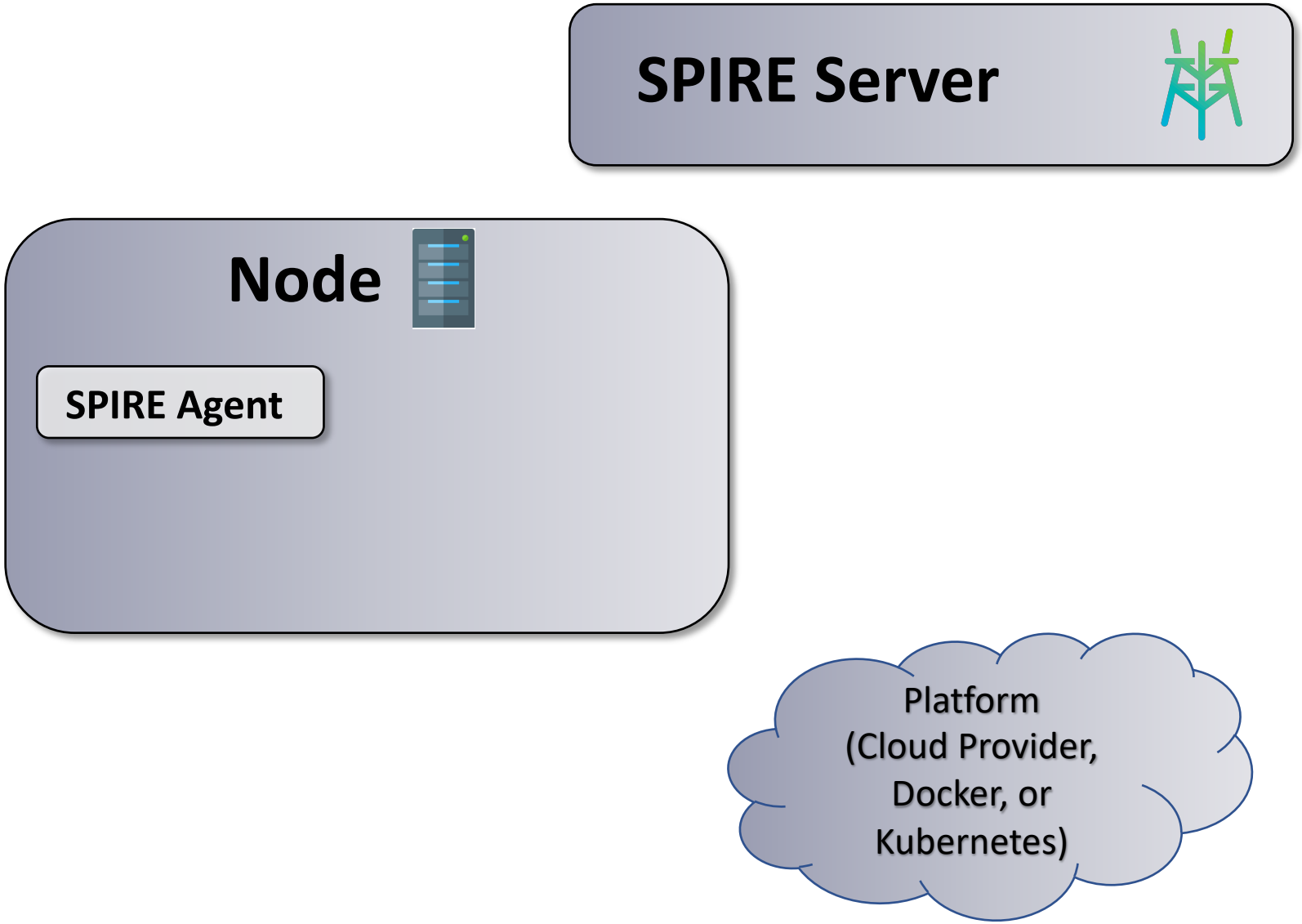
Network Service Mesh

SPIRE

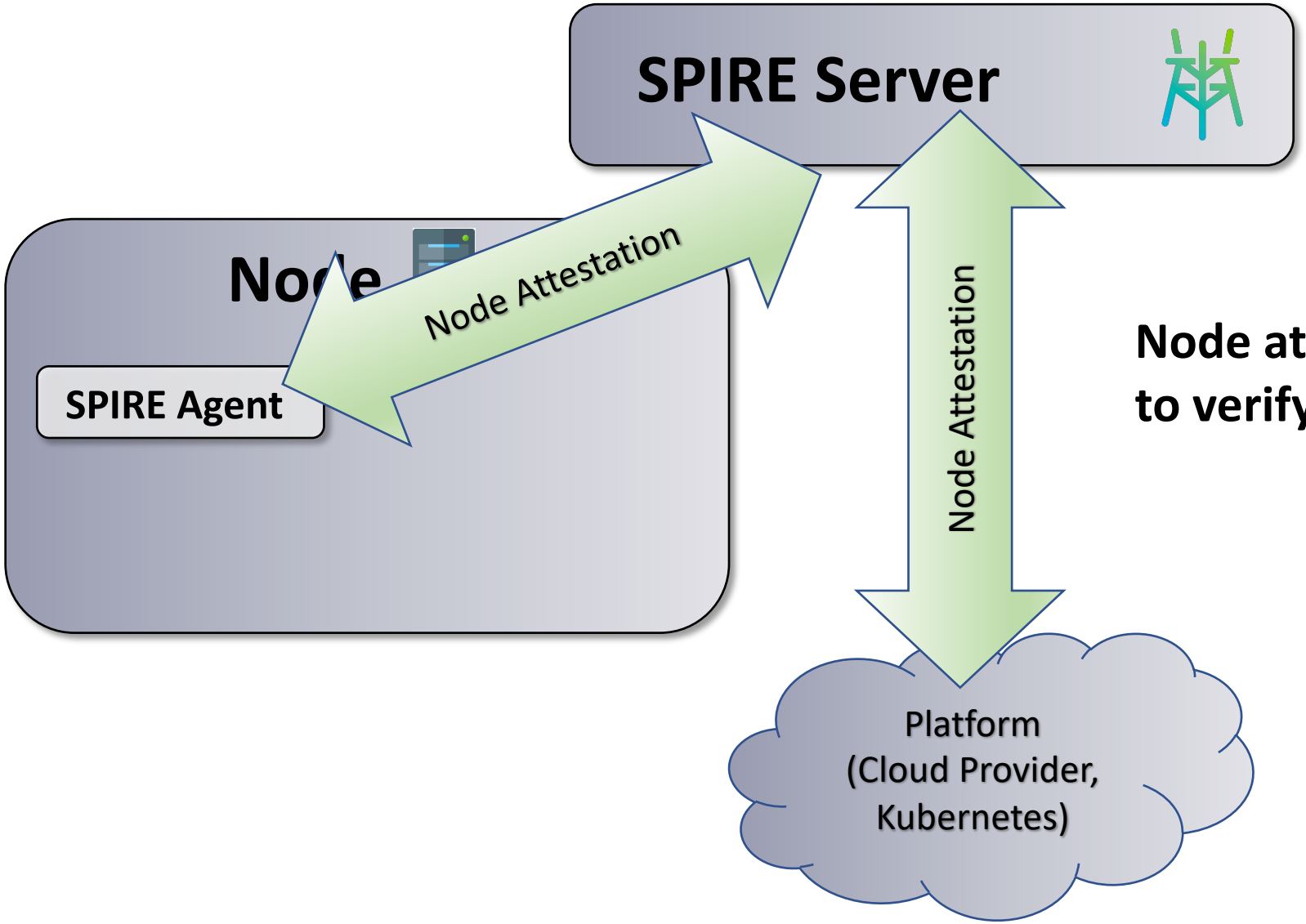
SPIRE (our implementation)



SPIRE (our implementation)

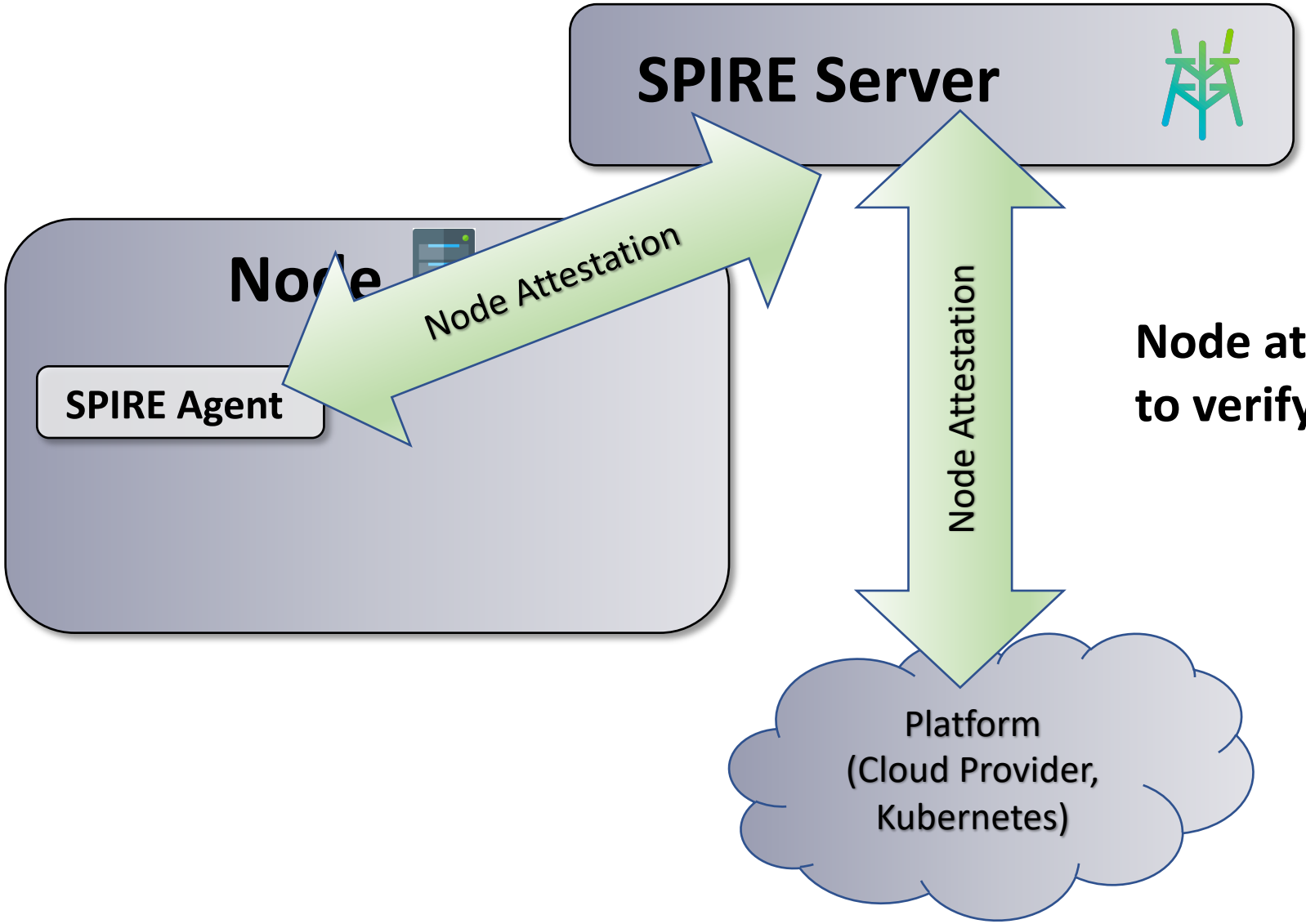


SPIRE (our implementation)



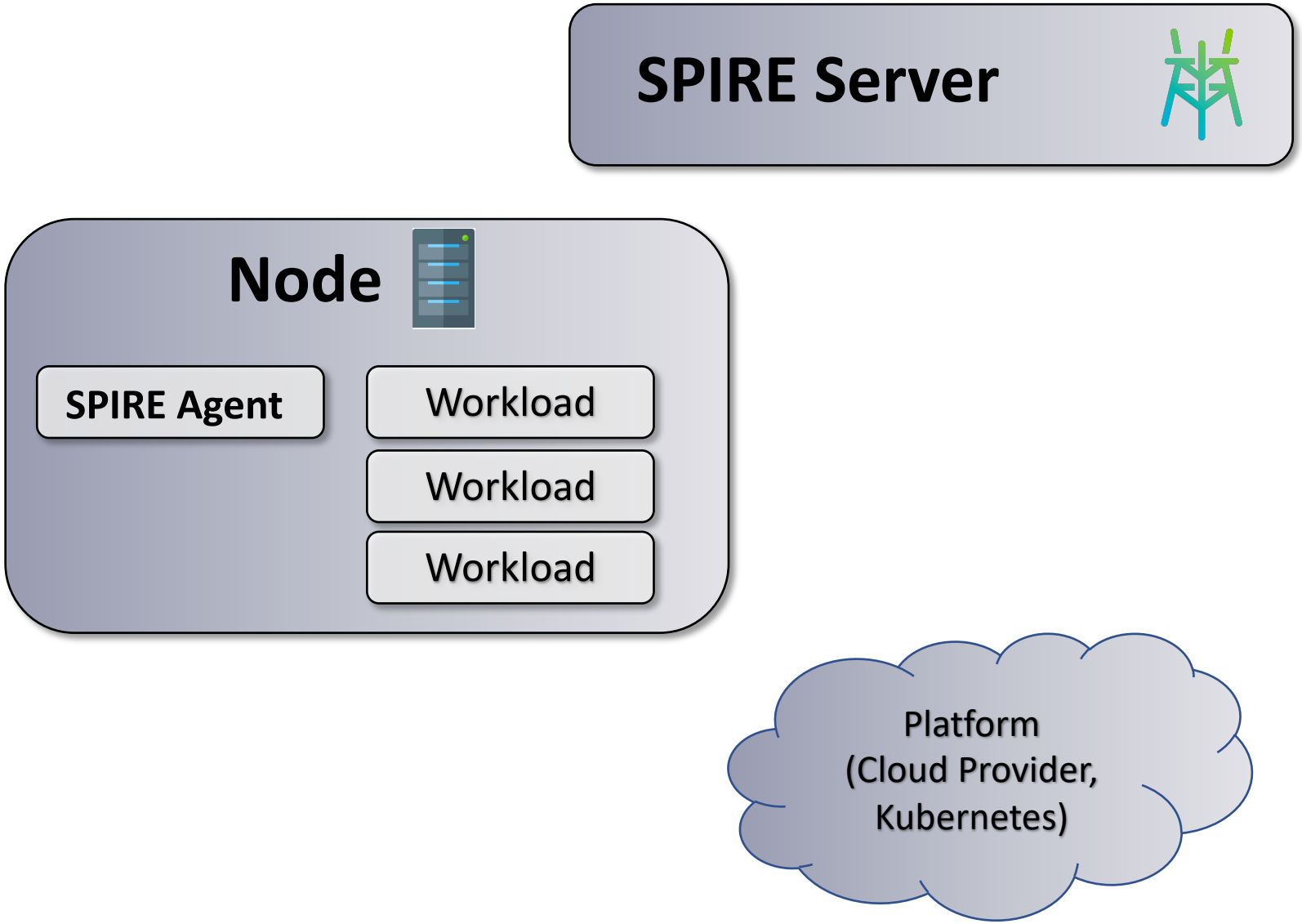
Node attestation uses the platform to verify the identity of the node

SPIRE (our implementation)

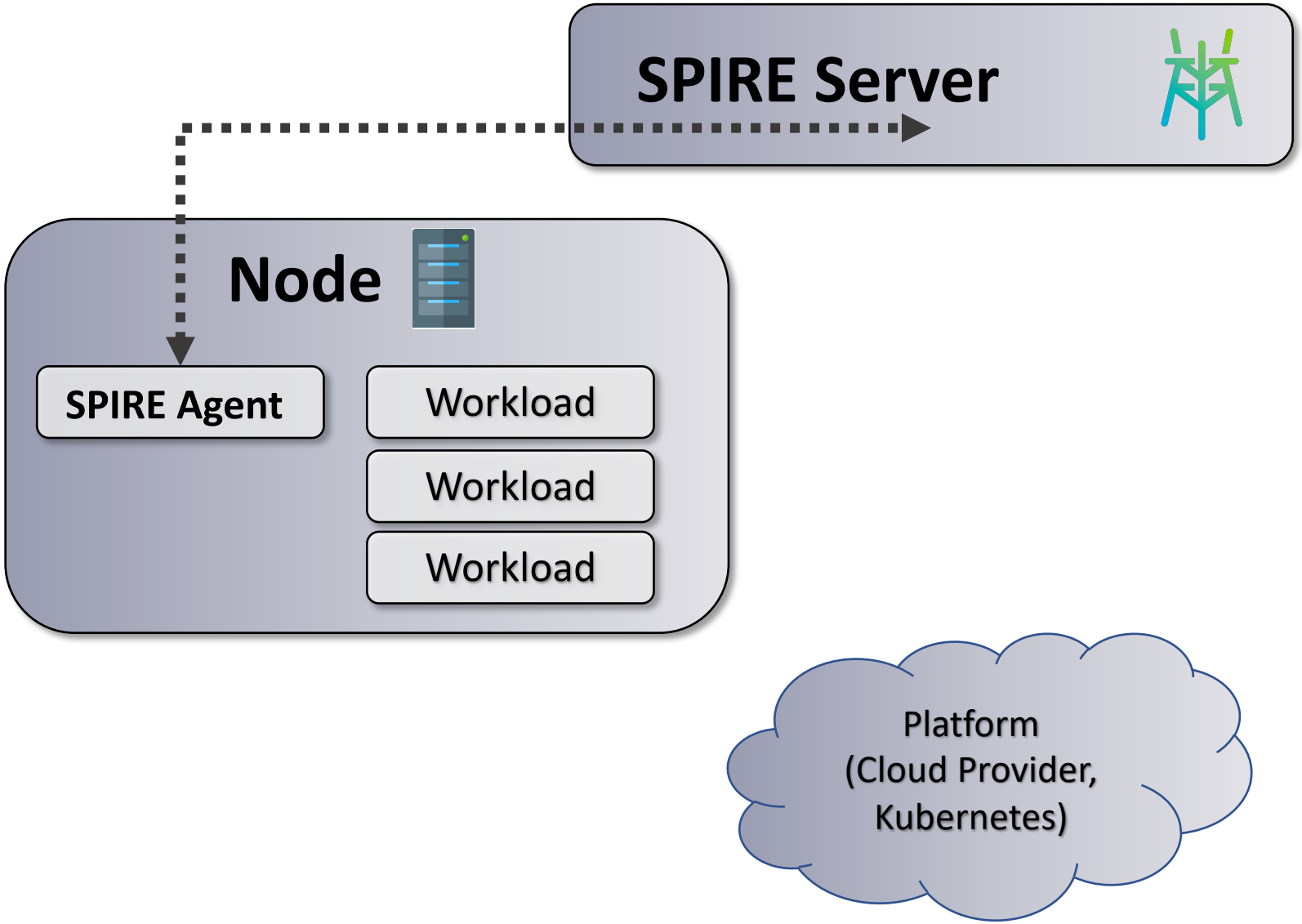


Node attestation uses the platform to verify the identity of the node

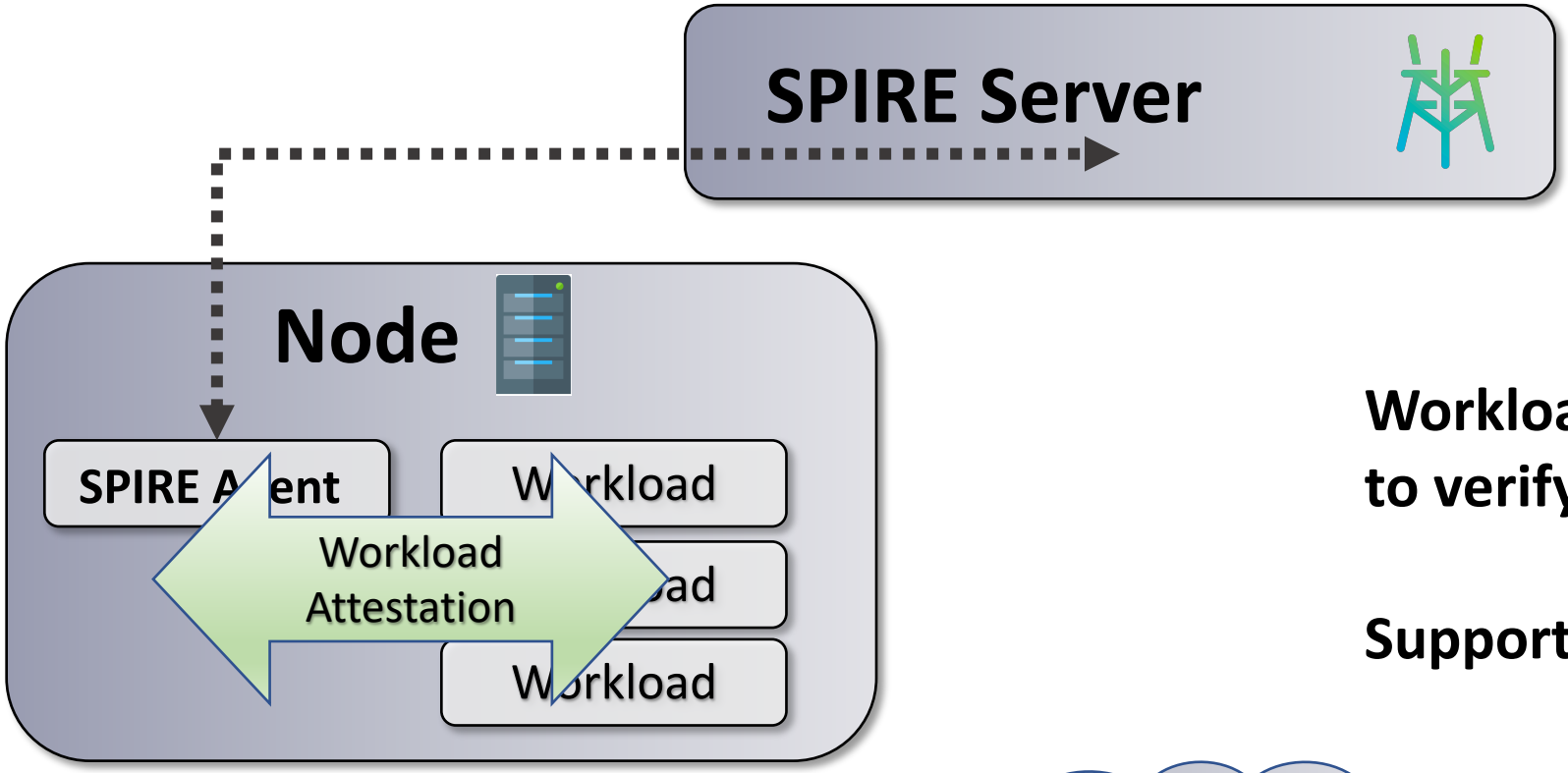
SPIRE (our implementation)



SPIRE (our implementation)



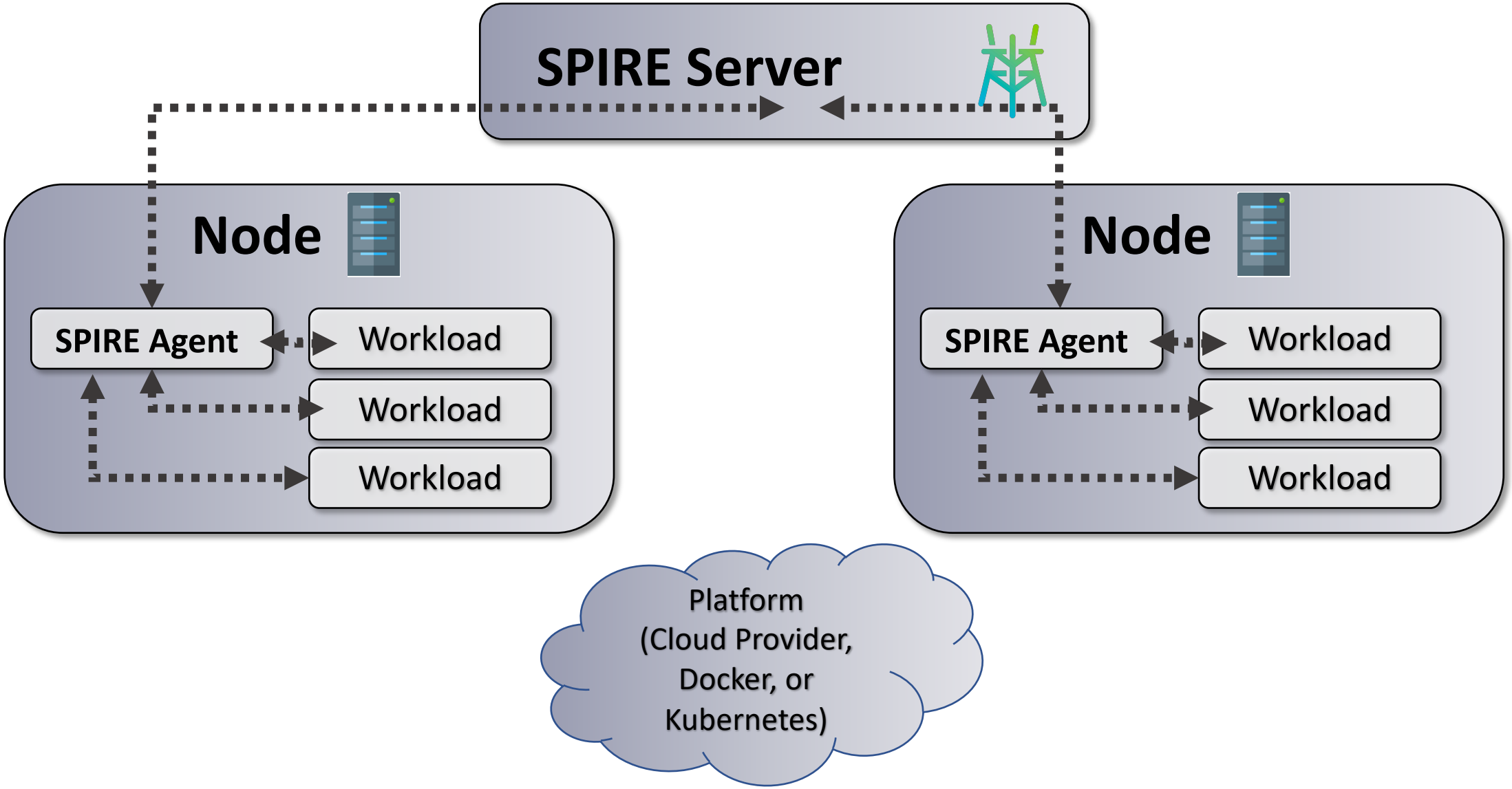
SPIRE (our implementation)



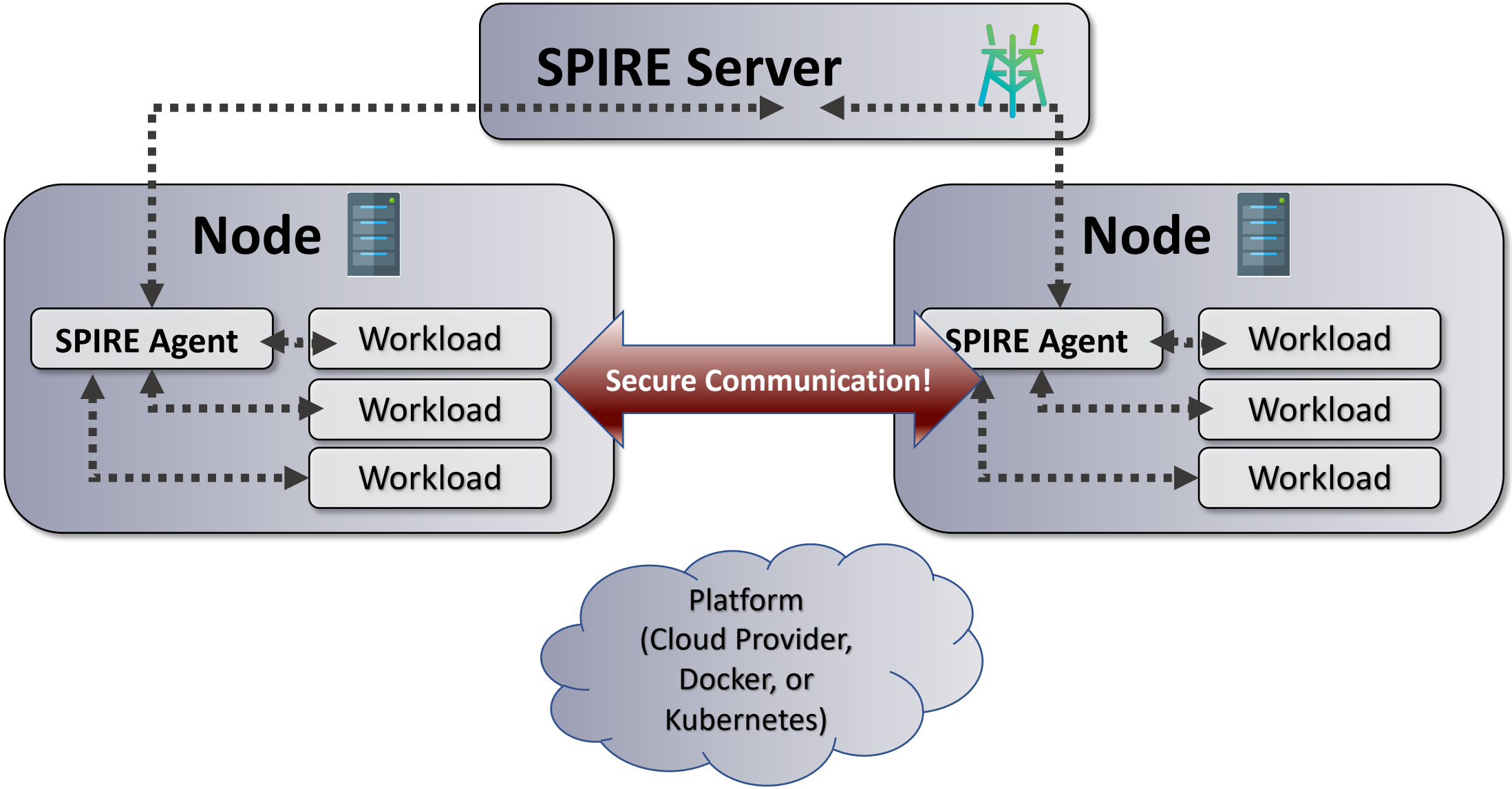
Workload attestation uses the kernel to verify the identity of the process

Supports Linux, BSD, Kubelet

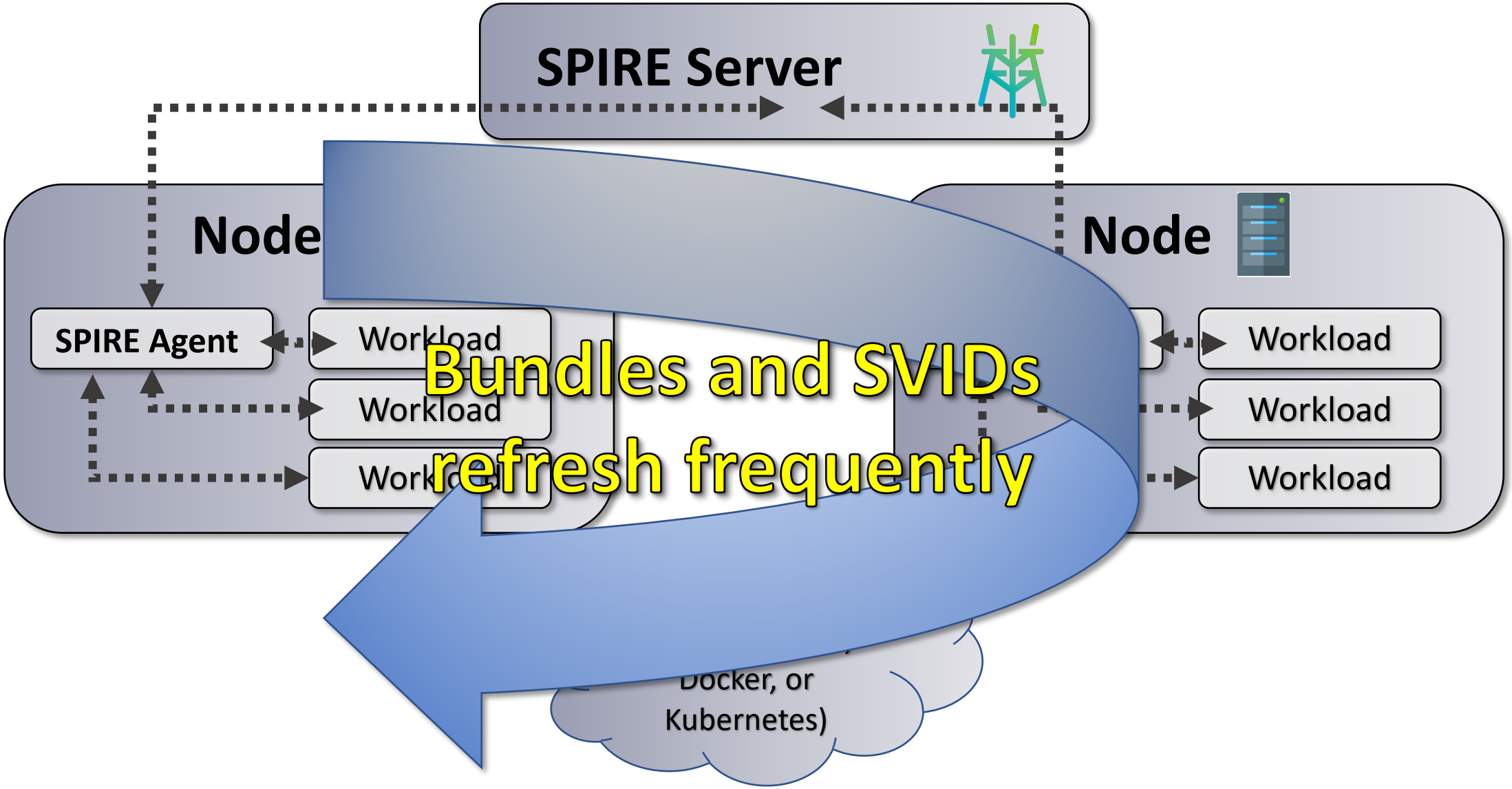
SPIRE (our implementation)



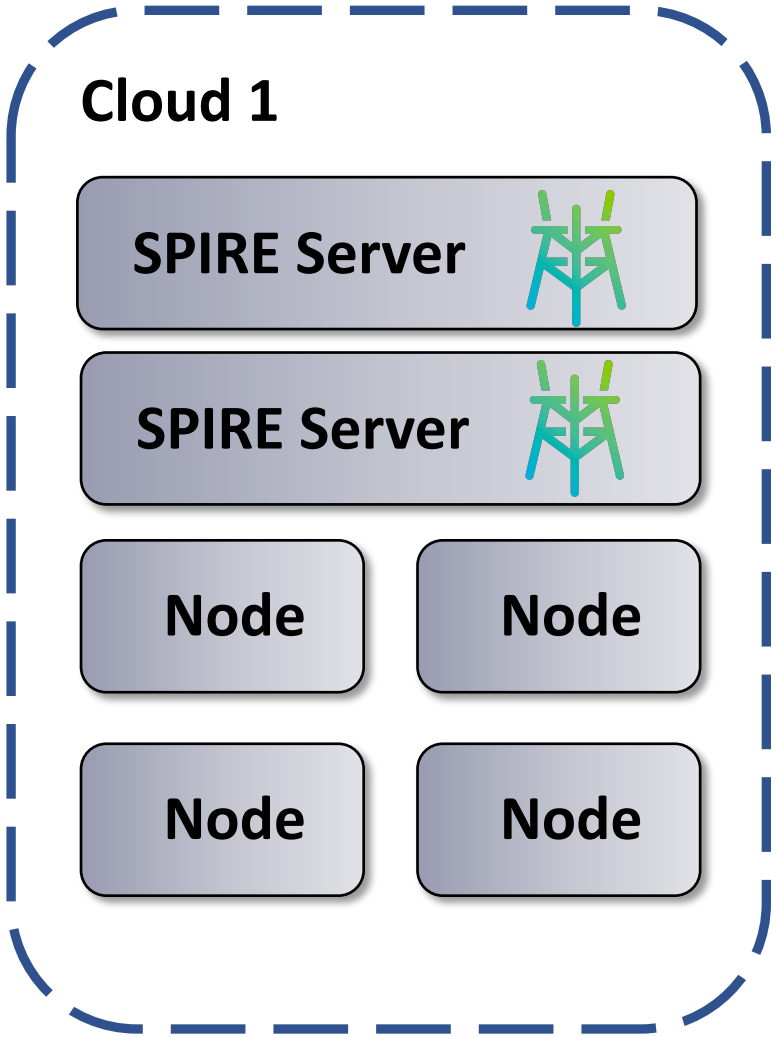
SPIRE (our implementation)



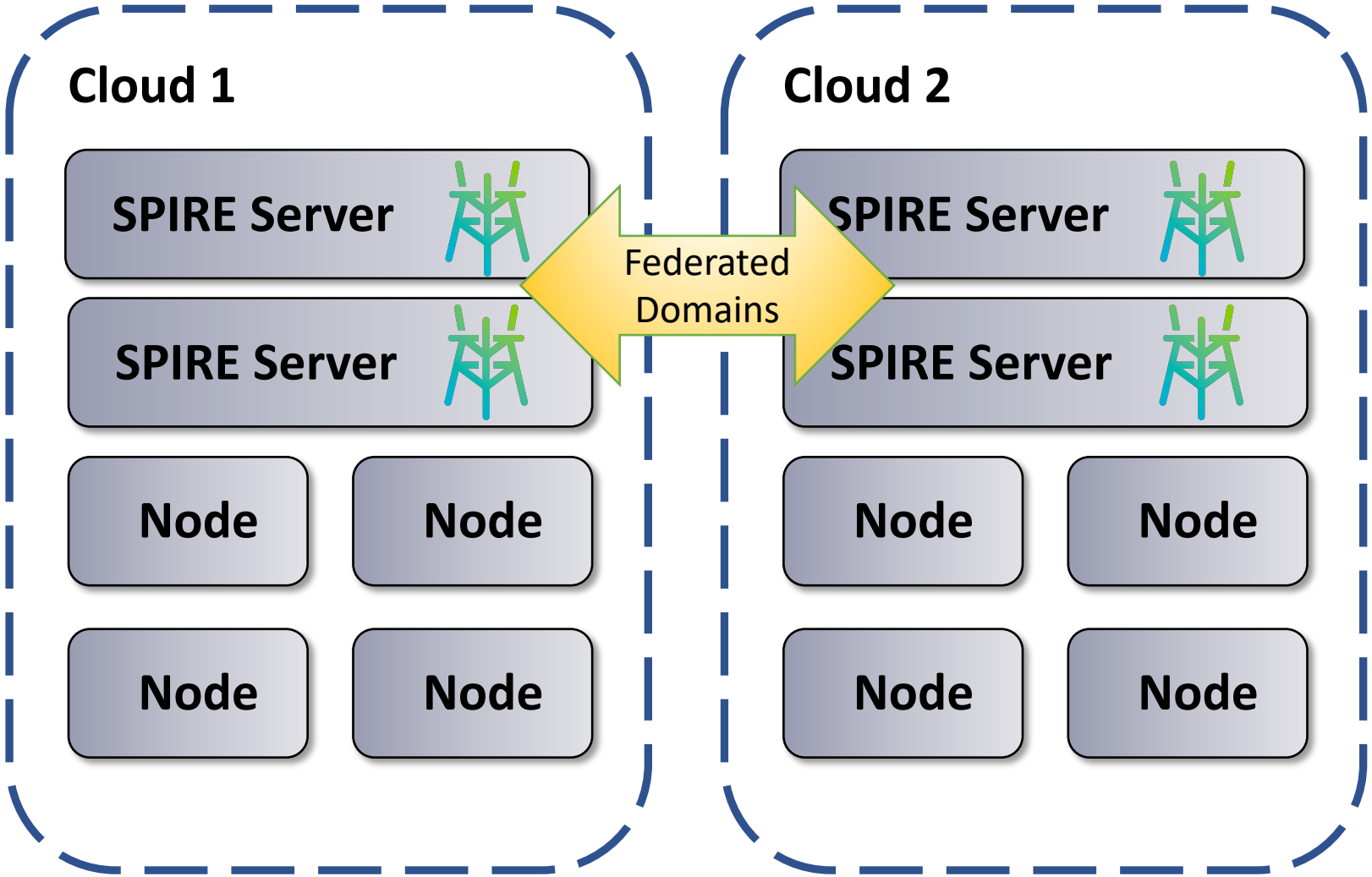
SPIRE (our implementation)



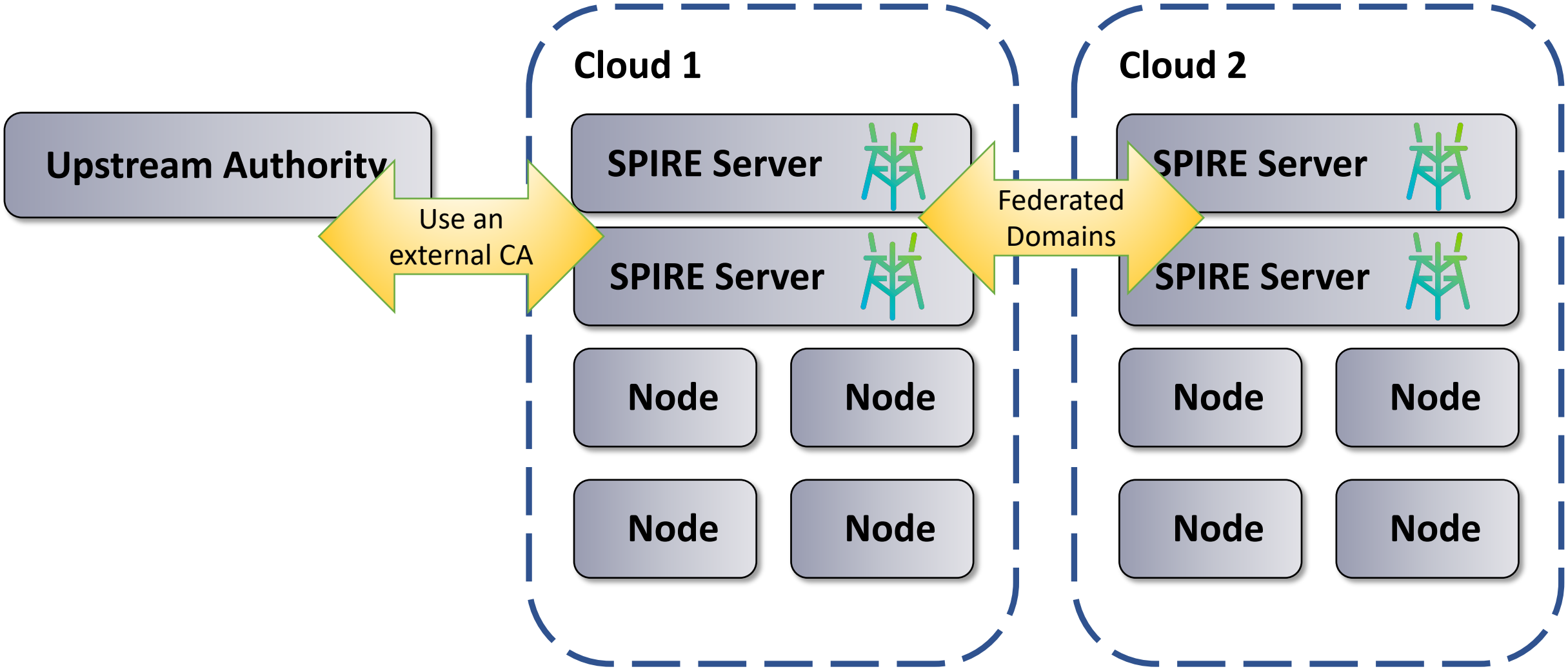
SPIRE (our implementation)



SPIRE (our implementation)



SPIRE (our implementation)



SPIRE (our implementation)

- Why do I need one SPIRE Agent per node?

SPIRE (our implementation)

- Why do I need one SPIRE Agent per node?
- What if the SPIRE Agent is compromised?

SPIRE (our implementation)

- Why do I need one SPIRE Agent per node?
- What if the SPIRE Agent is compromised?
- What about my old workload that I don't want to change?

SPIRE (our implementation)

- Why do I need one SPIRE Agent per node?
- What if the SPIRE Agent is compromised?
- What about my old workload that I don't want to change?
- How do I specify which workloads get which SPIFFE IDs?

Registration Entries

Node Registration Entry

If a node has:

The aws tag “database”

Then give it the SPIFFE ID:

spiffe://mycompany/database

SPIRE (our implementation)

Registration Entries

Node Registration Entry

If a node has:

The aws tag “database”

Then give it the SPIFFE ID:

spiffe://mycompany/database

Workload Registration Entry

If a workload has:

The parent ID spiffe://mycompany/database

The UID “payments”

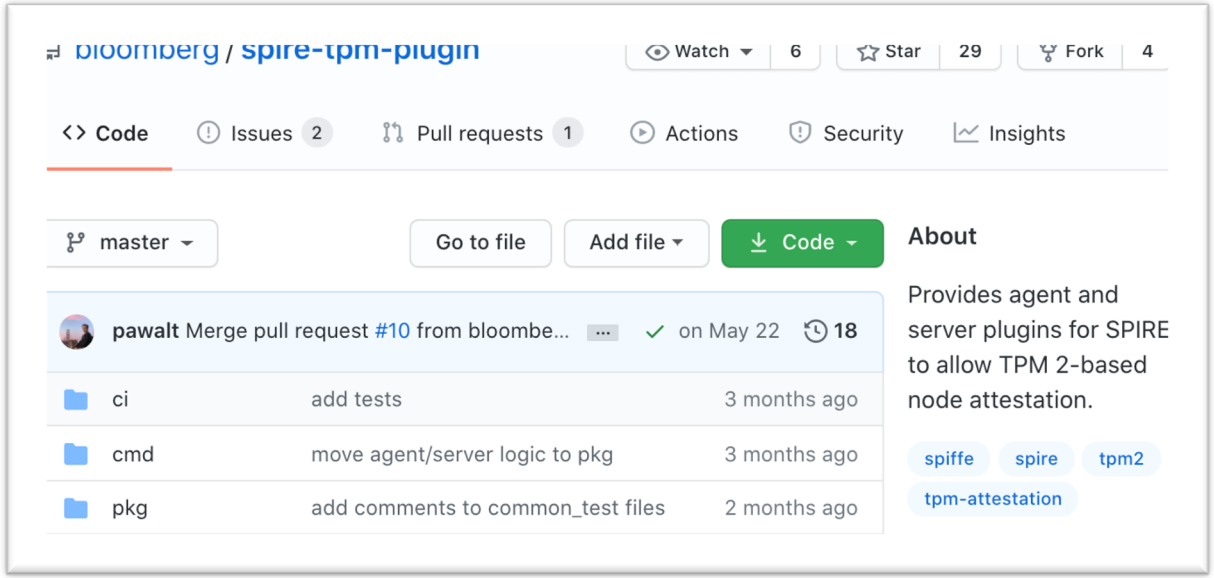
Then give it the SPIFFE ID:

spiffe://mycompany/database/payments

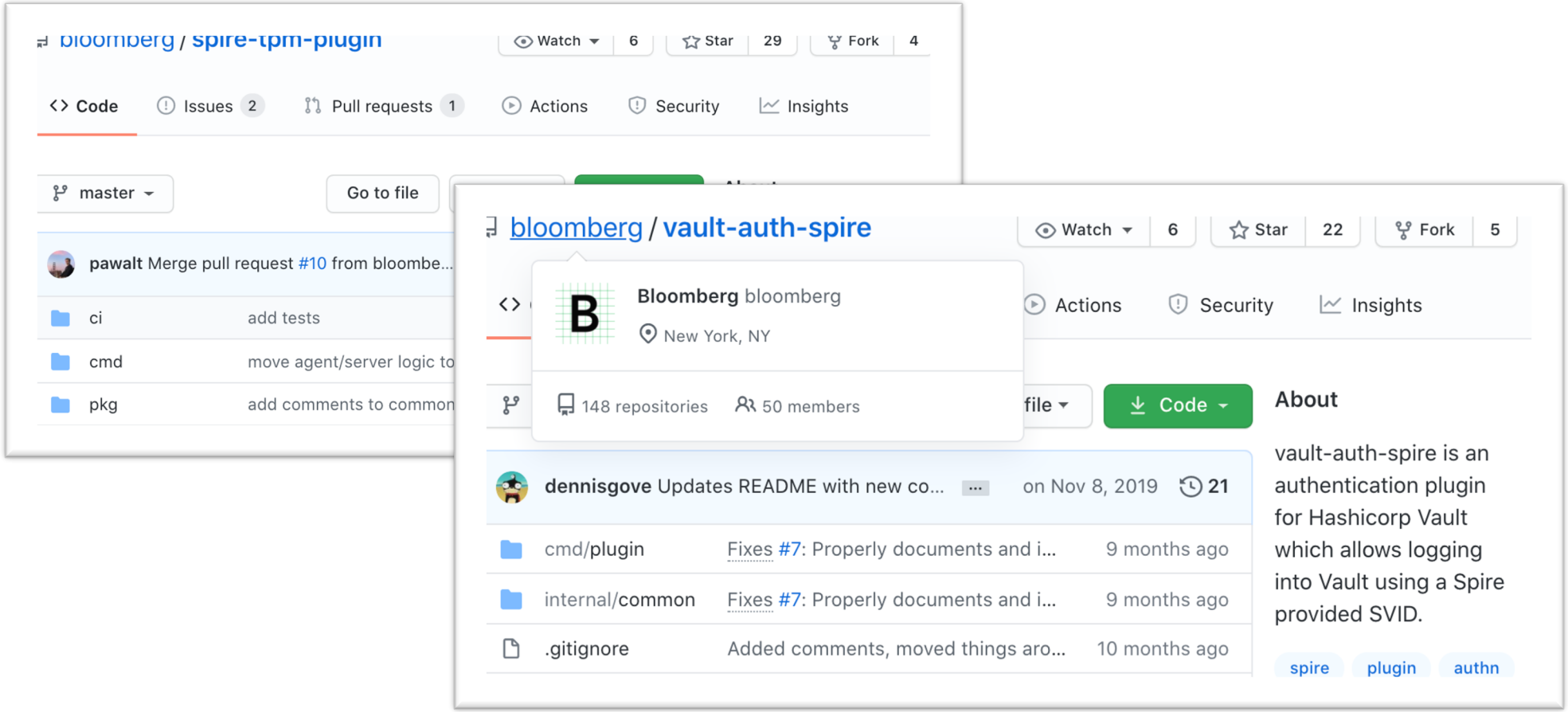
SPIRE (our implementation)

- Why do I need one SPIRE Agent per node?
- What if the SPIRE Agent is compromised?
- What about my old workload that I don’t want to change?
- How do I specify which workloads get which SPIFFE IDs?
- What if the SPIRE Server goes down?

SPIRE Ecosystem

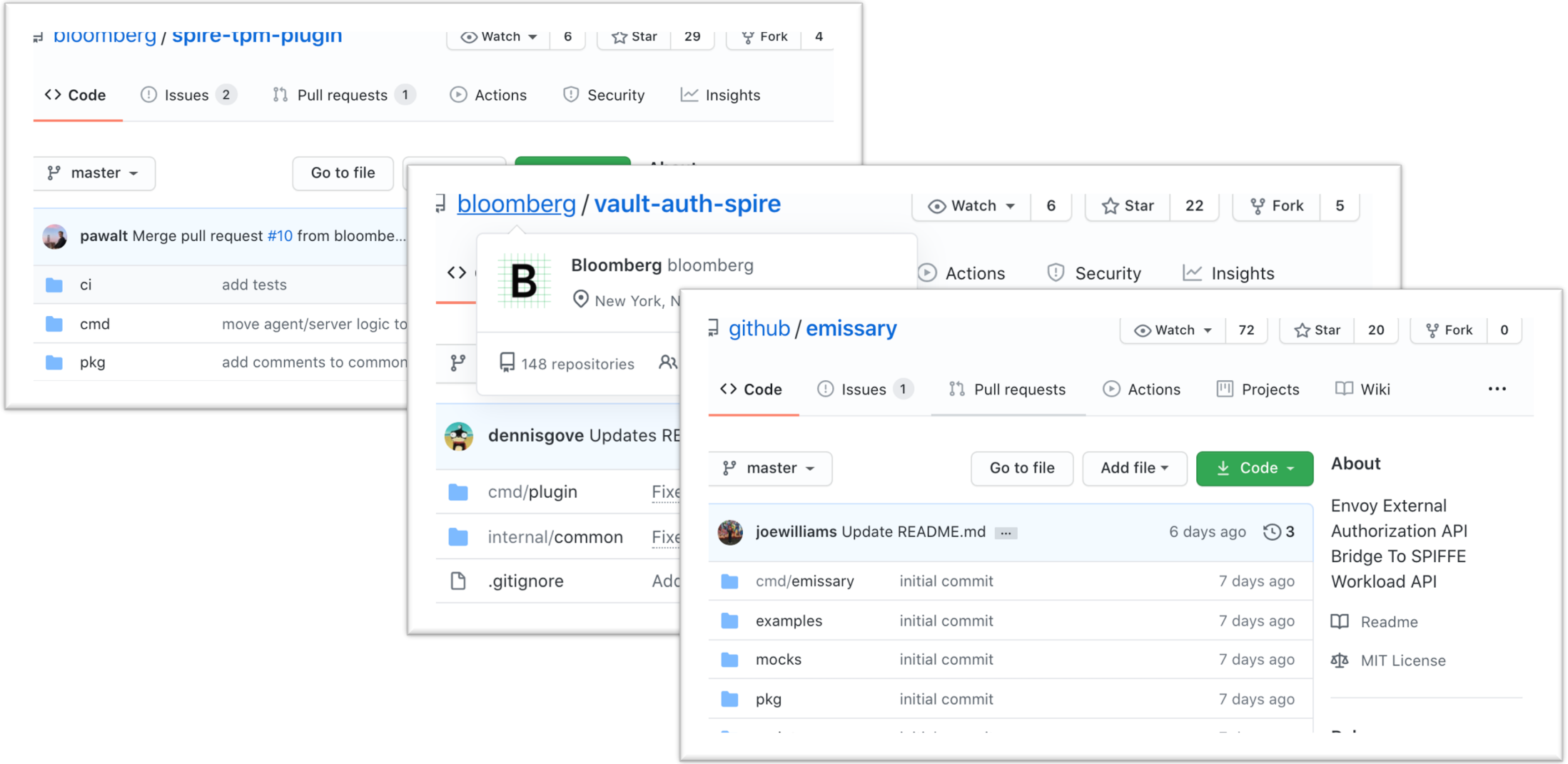


SPIRE Ecosystem



SPIRE Ecosystem

KubeCon CloudNativeCon Europe 2020 Virtual



Next Steps

Who's Using SPIFFE?

Anyone using Istio, Kuma, NSM, Consul Connect, Hamlet, or several other service mesh technologies is using at least part of SPIFFE!

Who's Using SPIRE?

ByteDance (TikTok)
Square
Bloomberg
Anthem (Health Insurance)
Uber
GitHub
Stripe
TransferWise

Others we don't know about….

Next Steps

KubeCon CloudNativeCon Europe 2020 Virtual

Learn More:
Find out more at spiffe.io
Regular Community Days
Join the SPIFFE Slack: slack.spiffe.io

More Talks:
Evan Gilman – Introduction to SPIFFE and SPIRE
Andrew Harding – Running SPIRE In Large Scale, Enterprise-Grade Environments
Building Zero Trust based Authentication in Healthcare with SPIRE

SPIRE Deep Dives at HPE Discover

Contribute!

Why Service Identity

Service Identity Approaches

SPIFFE (the standard)

SPIRE (the implementation)

Next Steps

Next Steps

Thank You!

@d_feldman

dan.feldman@hpe.com

Time for Questions


KubeCon
CloudNativeCon
Europe 2020
Virtual
HELM
KEEP CLOUD NATIVE
CONNECTED