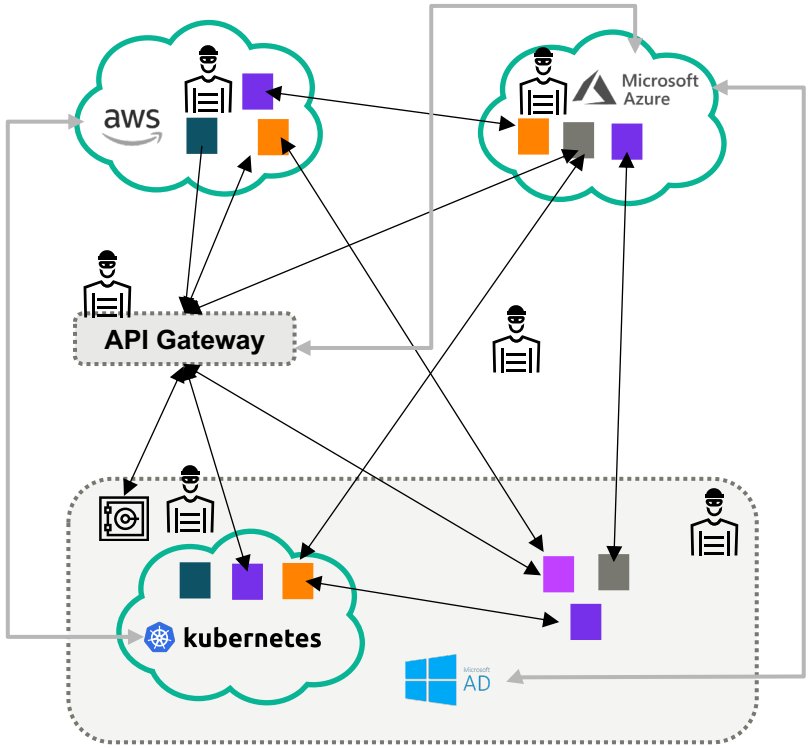
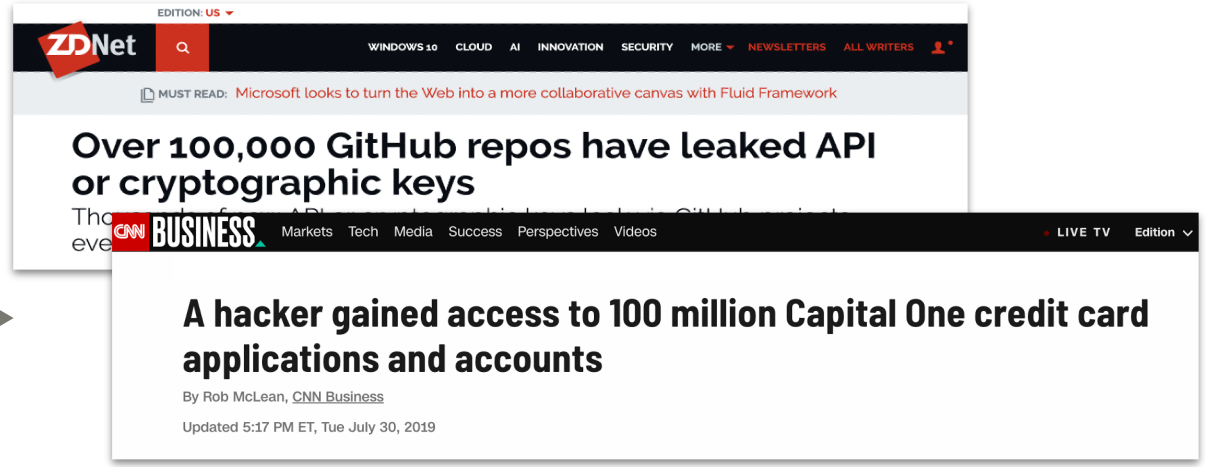
INTRODUCTION TO SPIFFE AND SPIRE (UHG)

Madhukesh Wali

CROSS-SERVICE COMMUNICATION IS EXPLODING

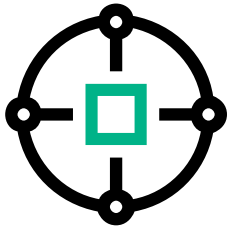
Increasing attack surface and risk of credential leakage across untrusted networks





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

EXISTING SOLUTIONS HAVE THEIR LIMITATIONS

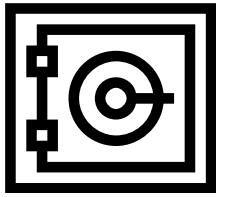


Network-based Access Tools

Assume the network is safe, and are unsuitable for dynamic environments

Legacy Protocols

Cannot be easily extended to the cloud or container-based environments.



Secret Stores

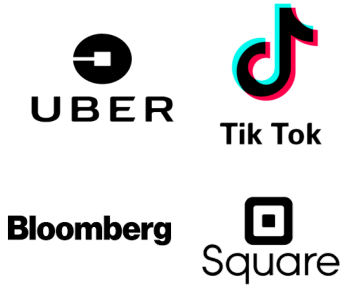
Require a secure introduction to services and create a “locus of privilege”

INTRODUCING SPIFFE AND SPIRE

Open-source specification and toolchain for service identity.

Part of Cloud Native Foundation

Integrated into various open-source projects.



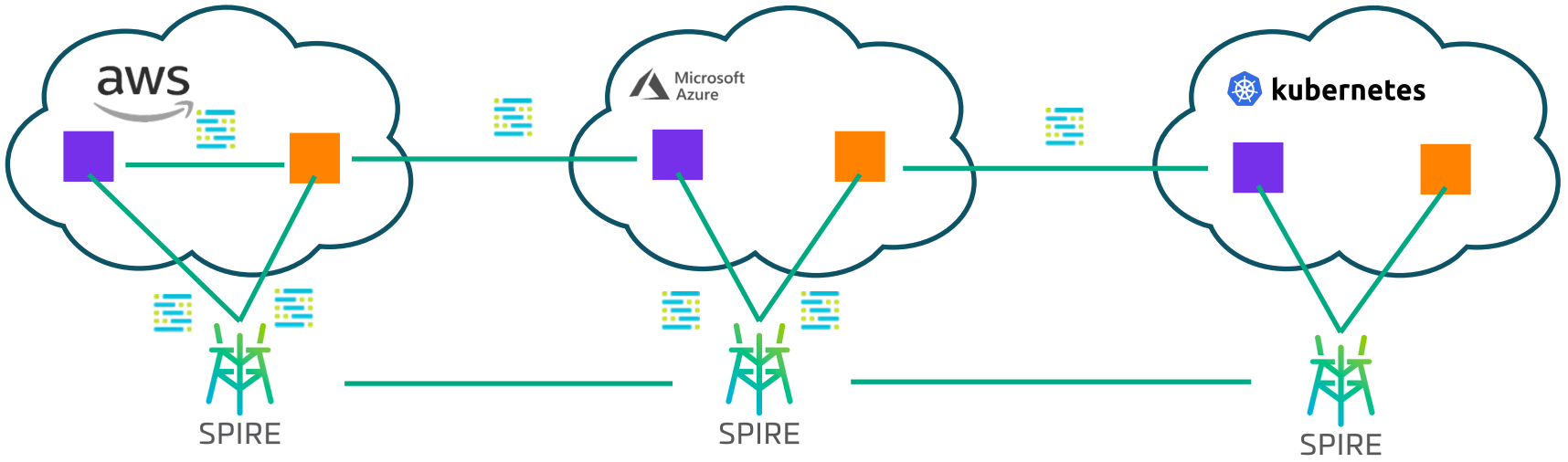
Extensive contributions by HPE and Fortune 2000 enterprises.

SPIFFE OVERVIEW

Key Attributes

SPIRE

Industry First, Cloud Native Service Identity Plane based on SPIFFE



Easily establish trust between services across platforms
without necessarily using secrets or network *security controls*

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?

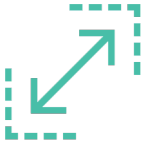
Can we affirm the integrity of the machine it runs upon?

- 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



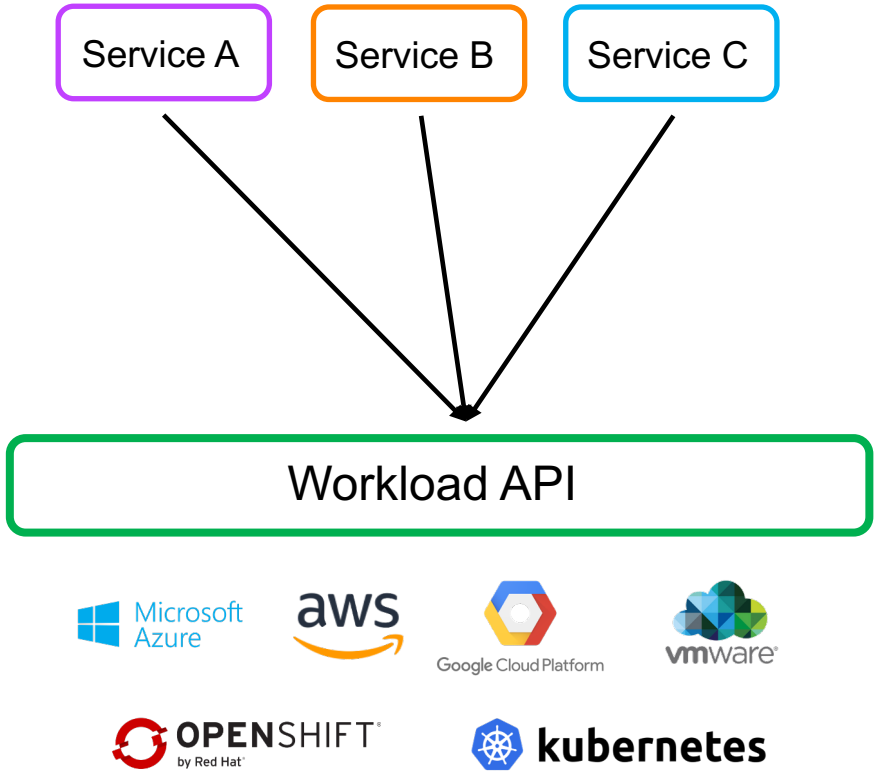
- Easily extends to identity providers, certificate authorities, and systems
- Designed for dynamic, distributed environments

BUILD A FOUNDATION FOR ZERO TRUST NETWORKS

Fine-grained authentication without necessarily relying on network security controls

PROVIDE “DIAL-TONE” AUTHENTICATION TO DEV TEAMS

Reduce operational complexity and risk by providing standard, frictionless authentication



HOW SPIFFE AND SPIRE WORK

SPIFFE AND SPIRE PROJECTS

github.com/spiffe/spiffe

A set of specifications that cover how a workload should retrieve and use it's identity.

- SPIFFE ID
- SPIFFE Verifiable Identity Documents (SVIDs)
- The SPIFFE Workload API

github.com/spiffe/spire

The SPIFFE Runtime Environment. Open-source software that implements the SPIFFE Workload API for a variety of platforms.

Apache 2.0 license. Independent governance. Highly extensible through plug-ins.

SPIFFE ID

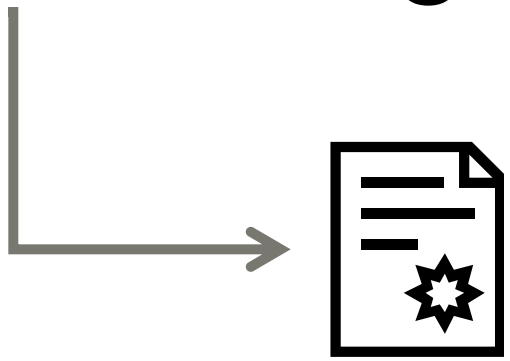
spiffe://acme.com/billing/payments

Trust Domain ***Workload Identifier***

SPIFFE VERIFIABLE IDENTITY DOCUMENT (SVID)

spiffe://acme.com/billing/payments

Typically short-lived



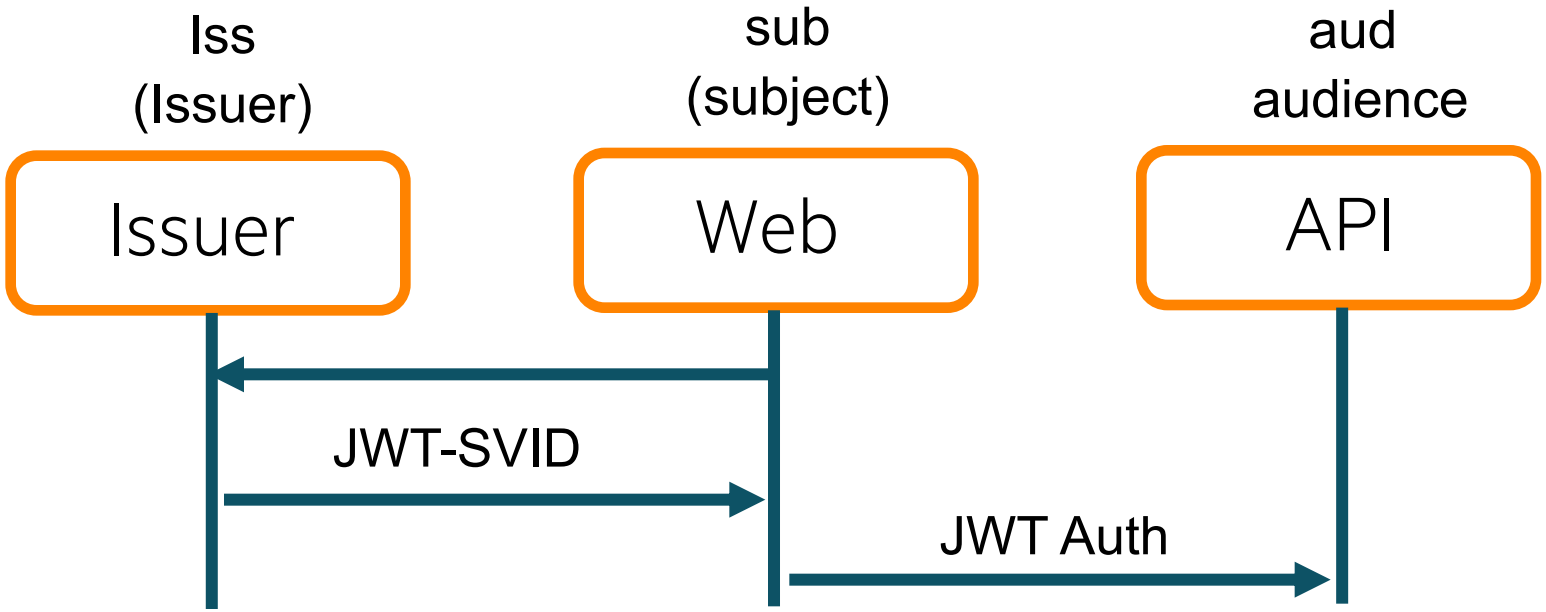
Today only one form of SVID (X509-SVID).
Other document types under consideration
(including JWT-SVID)

X.509 SVID

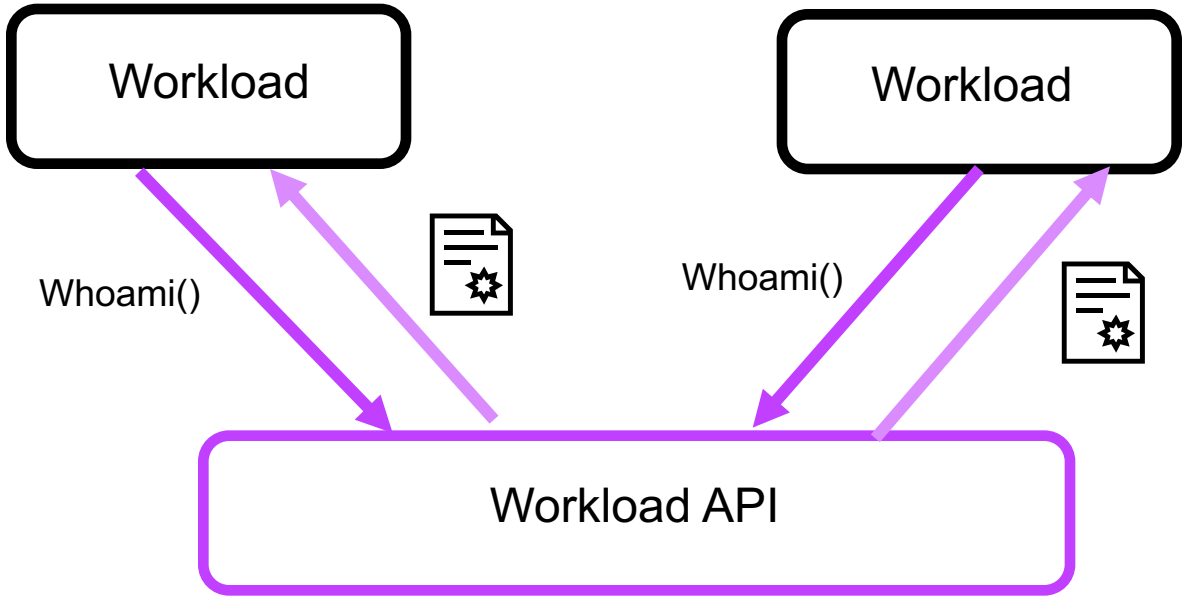
```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4 (0x4)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, O=SPIFFE
        Validity
            Not Before: Dec  1 15:30:54 2017 GMT
            Not After : Dec  1 16:31:04 2017 GMT
        Subject: C=US, O=SPIRE
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (521 bit)
                pub:
                    04:01:….
                ASN1 OID: secp521r1
                NIST CURVE: P-521
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Key Agreement
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Subject Alternative Name:
                URI:spiffe://example.org/host/workload
    Signature Algorithm: sha256WithRSAEncryption
```

JWT SVID

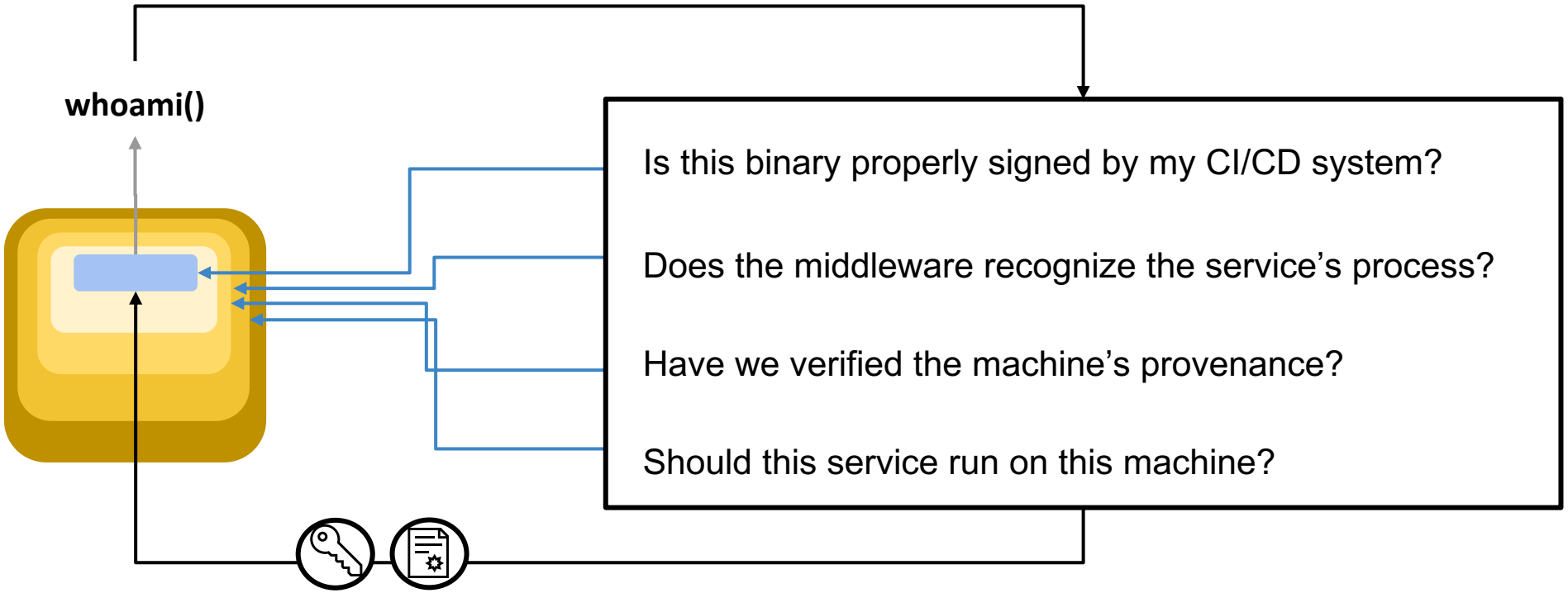
```
{
    “aud”:  “sidecar”,
    “exp”:   1551170050,
    “iat”:    1551169760,
    “sub”:    “spiffe://staging1.acme.com/payments/web"
}
```



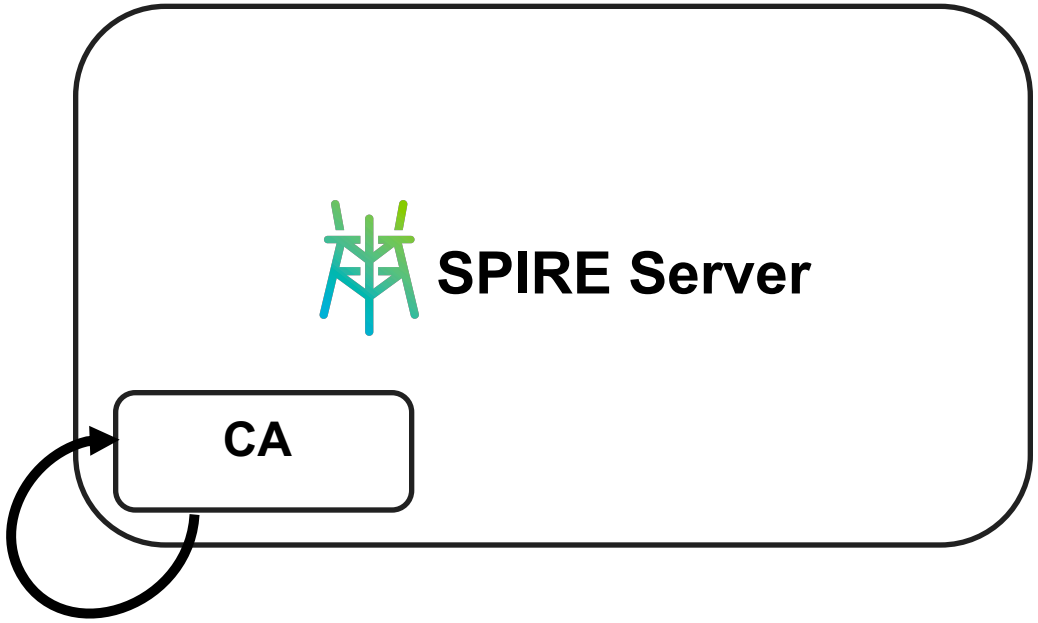
SPIFFE WORKLOAD API



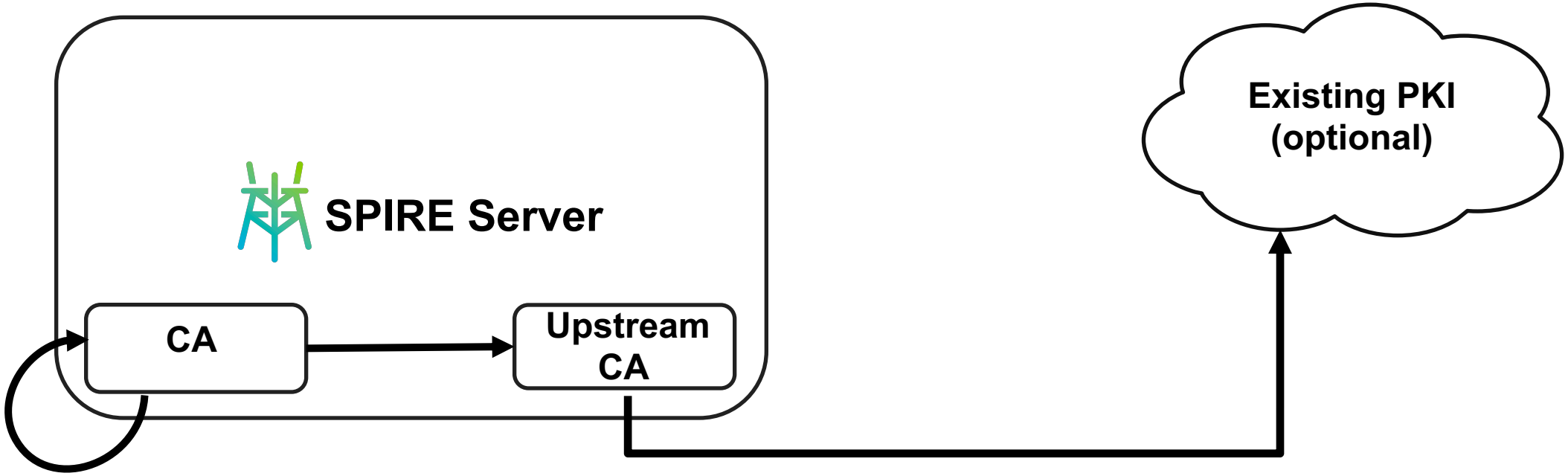
SPIRE *verifies* SPIFFE passports issued to software



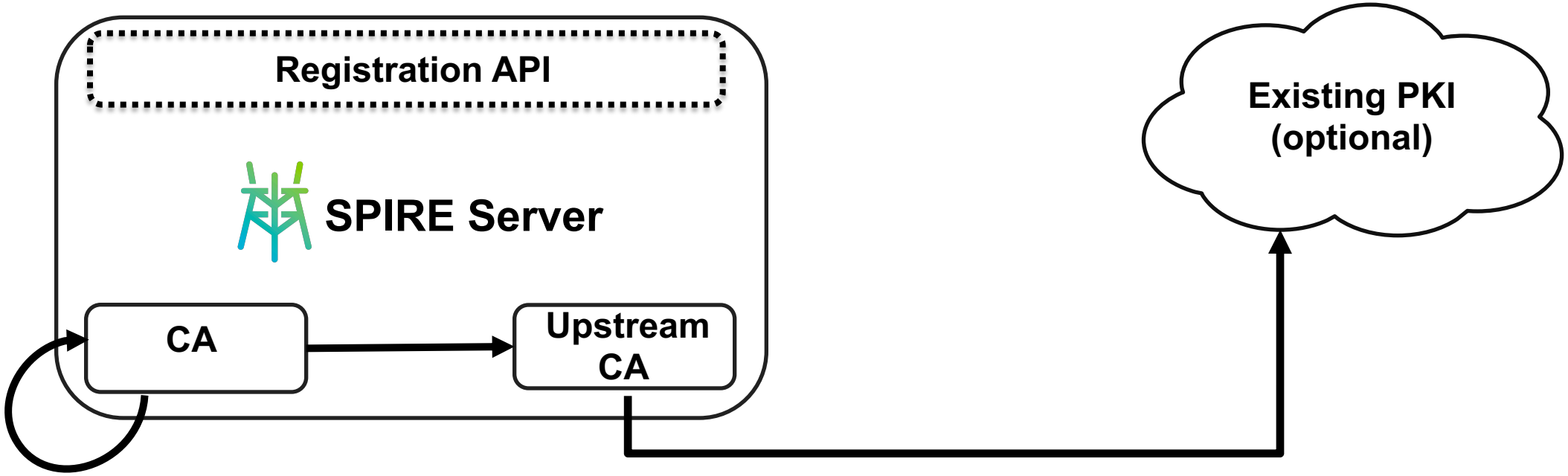
SPIRE Walkthrough



SPIRE Walkthrough



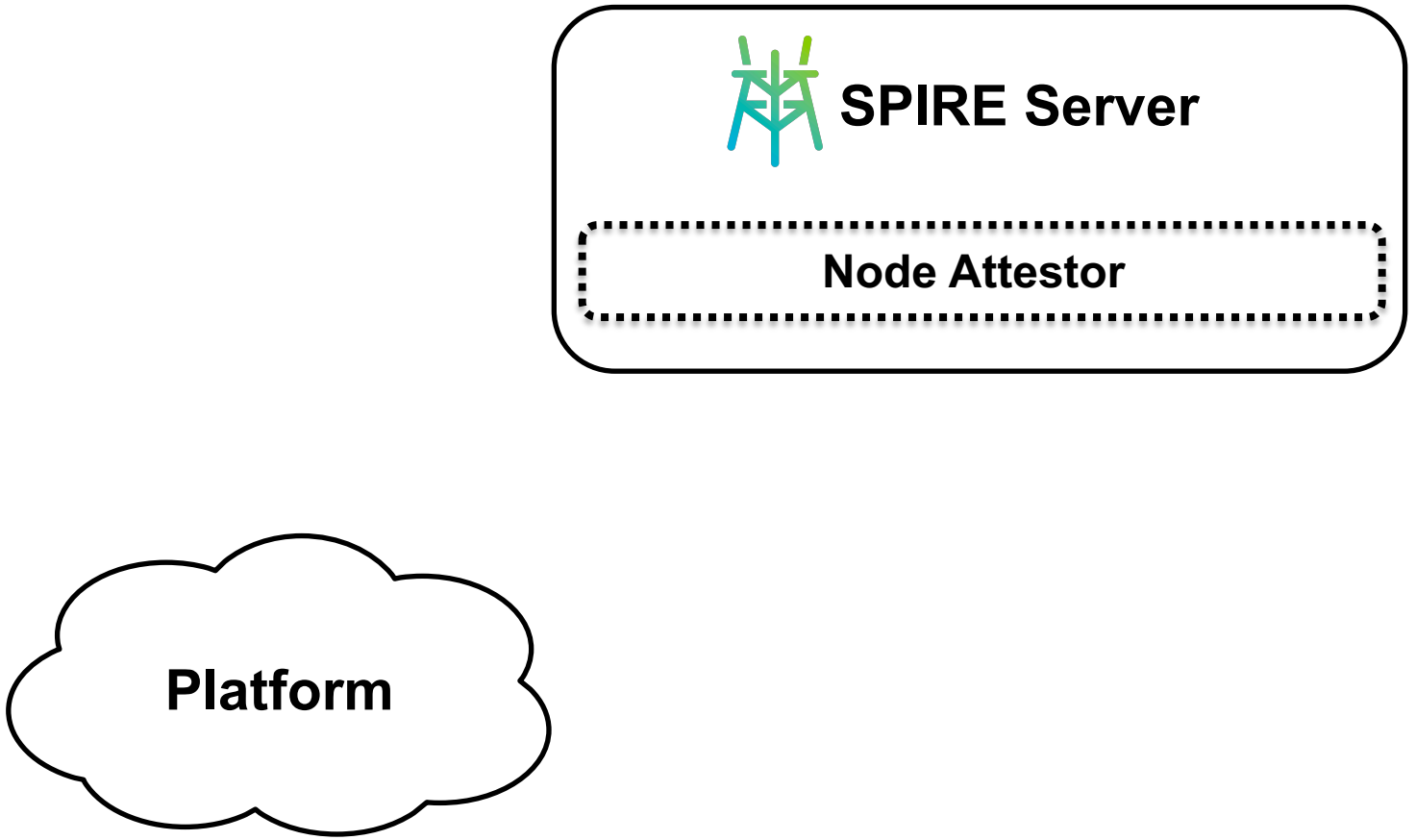
SPIRE Walkthrough



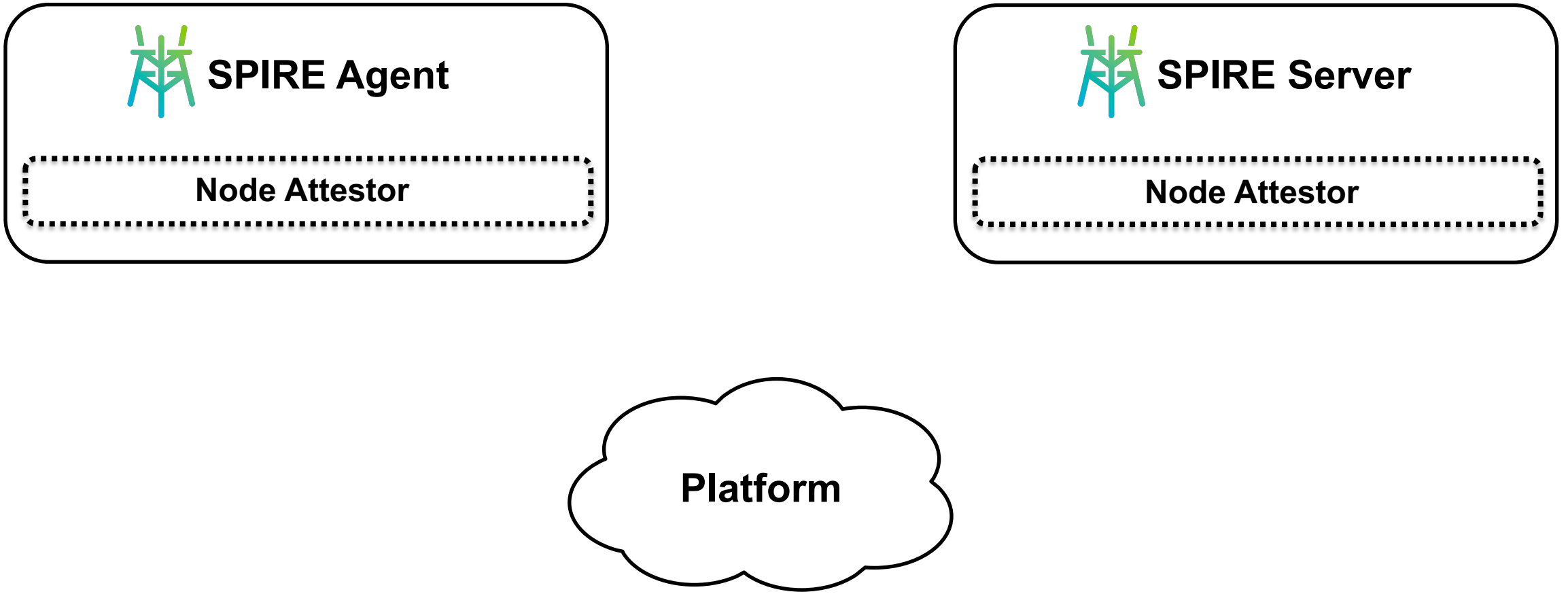
SPIRE Walkthrough

Parent ID:	spiffe://staging.acme.com/k8s/cluster/staging
Selector:	K8s:ns:payments
Selector:	K8s:sa:staging
Selector:	docker:image-id: 746b819f315e
SPIFFE ID:	spiffe://staging.acme.com/payments/web

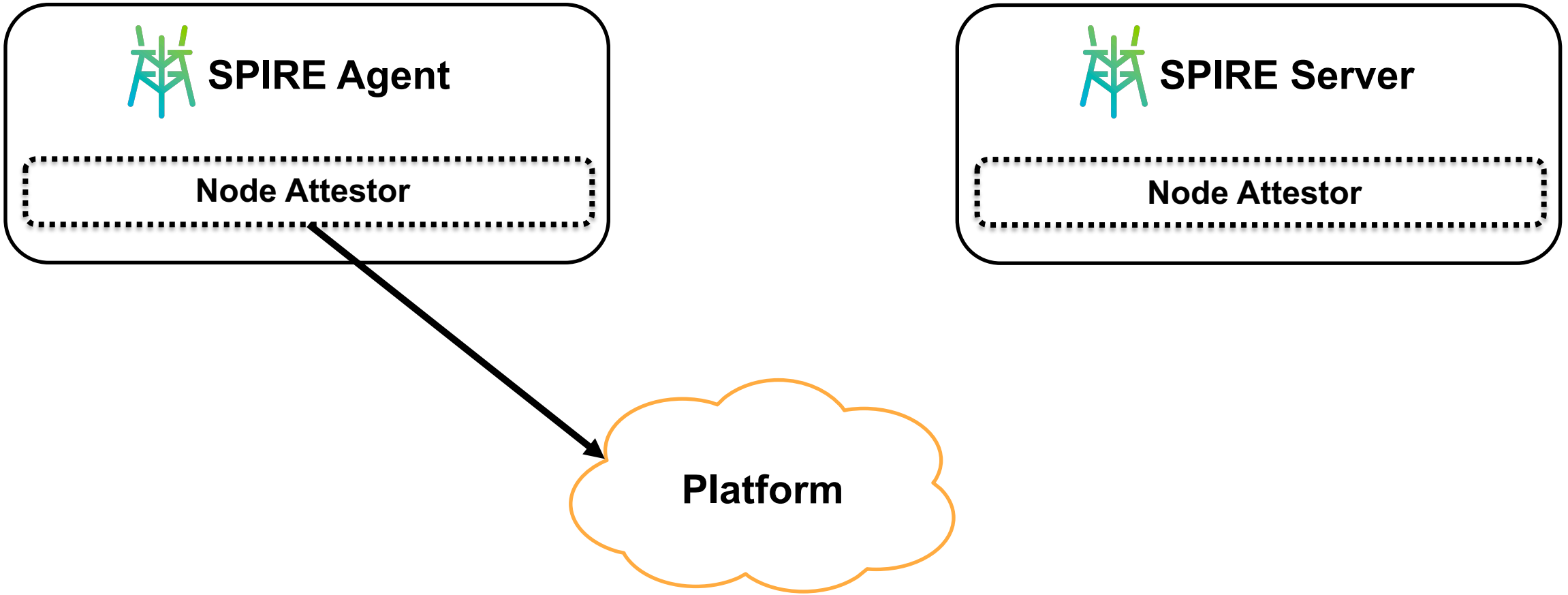
SPIRE Walkthrough



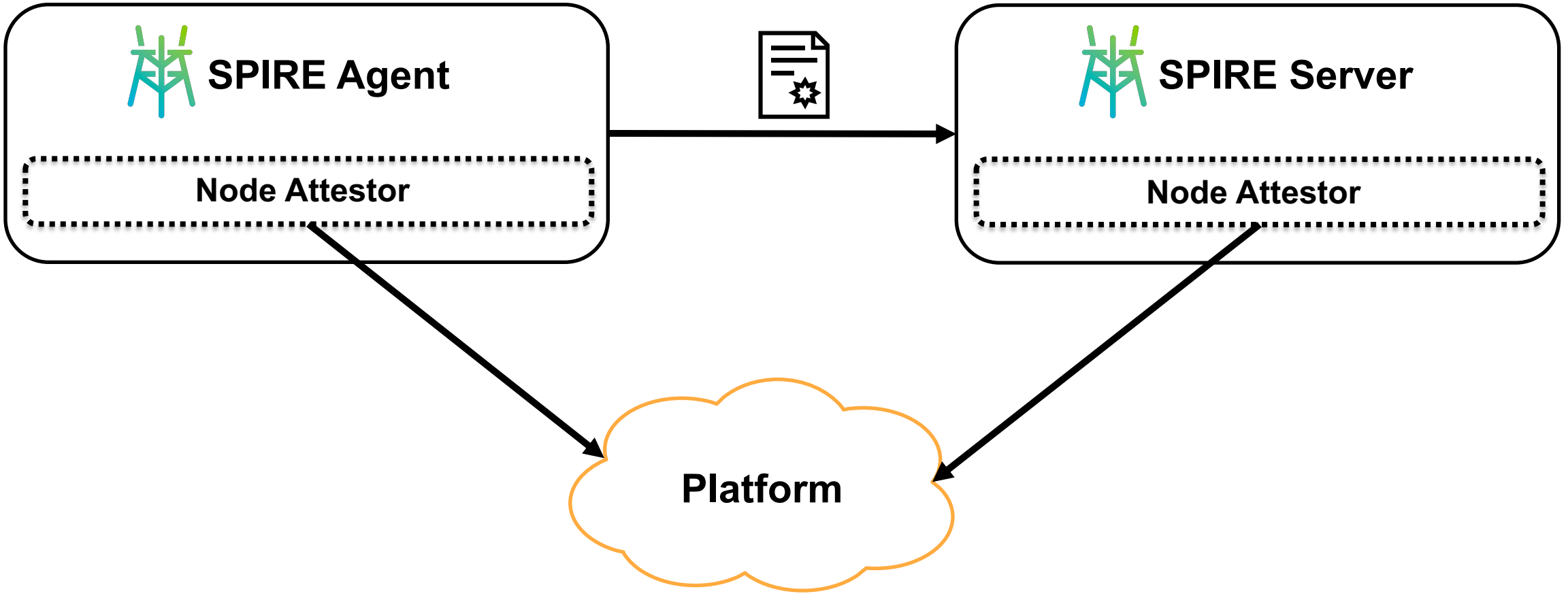
SPIRE Walkthrough



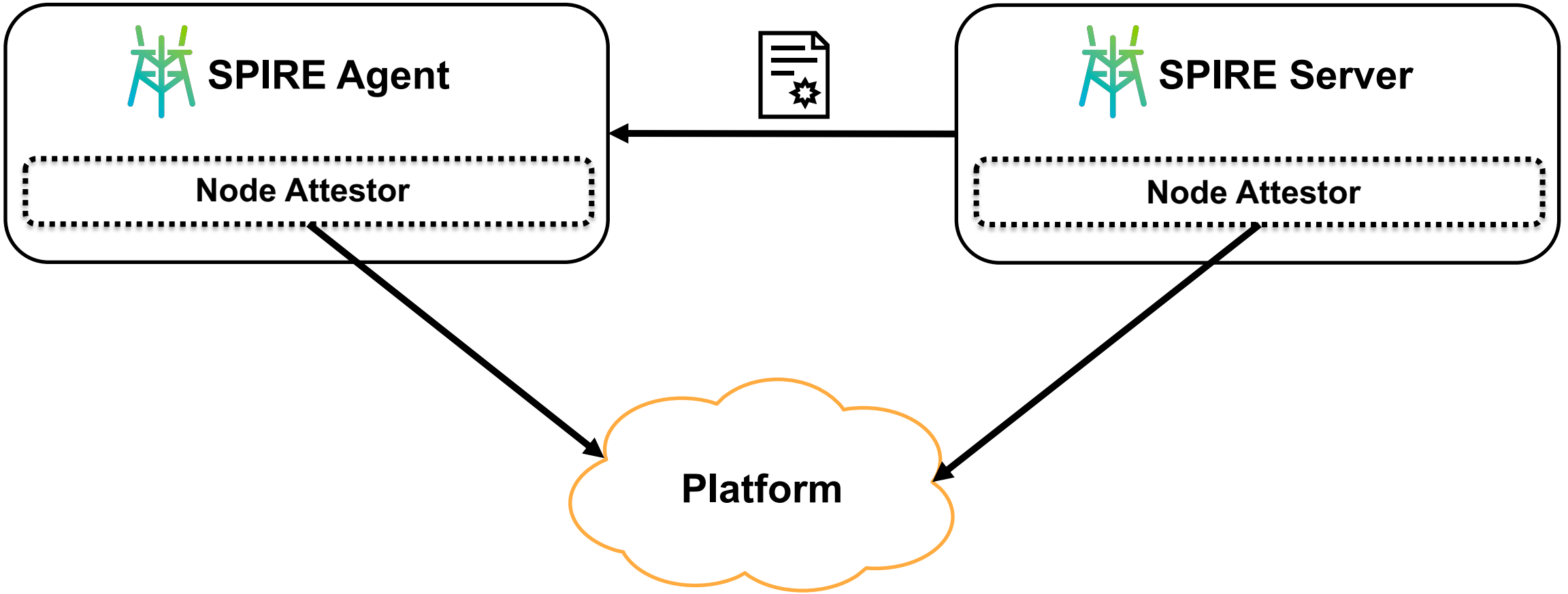
SPIRE Walkthrough



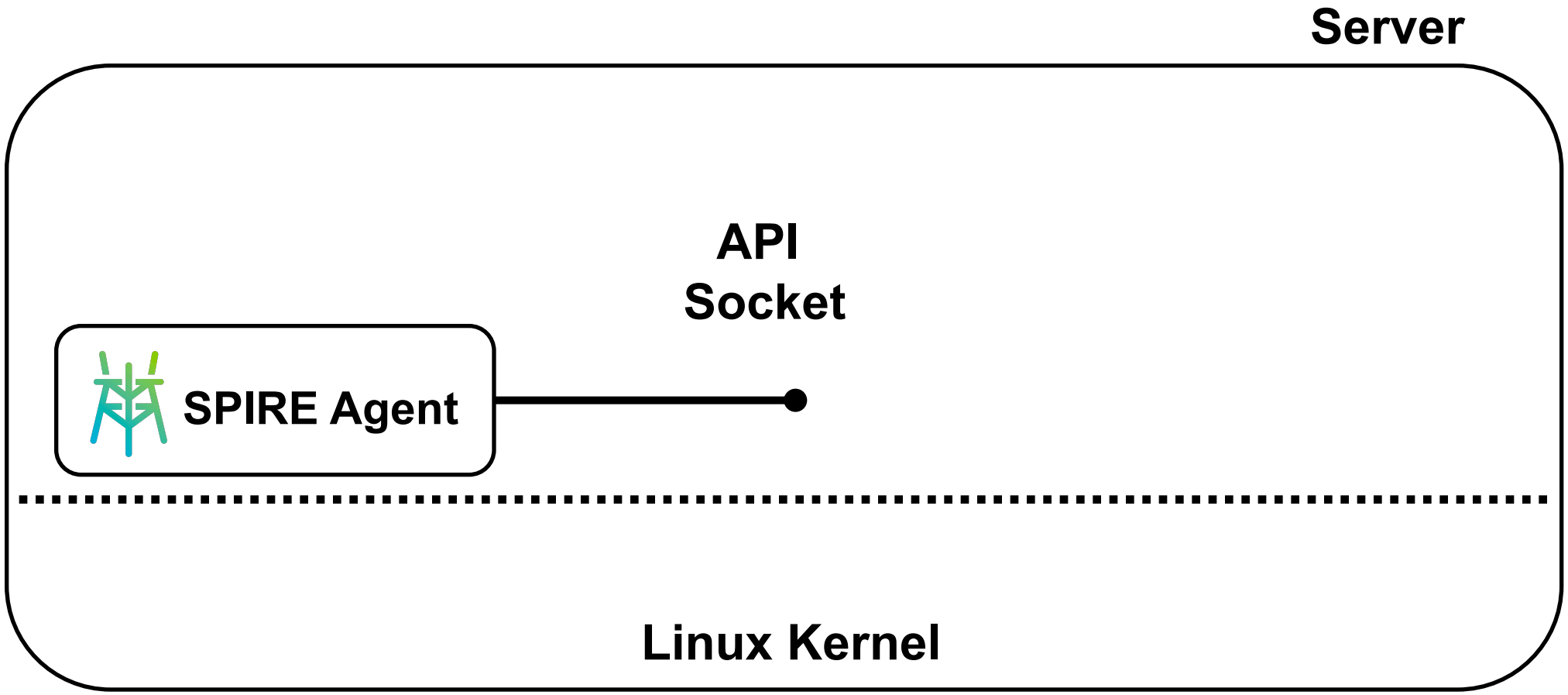
SPIRE Walkthrough



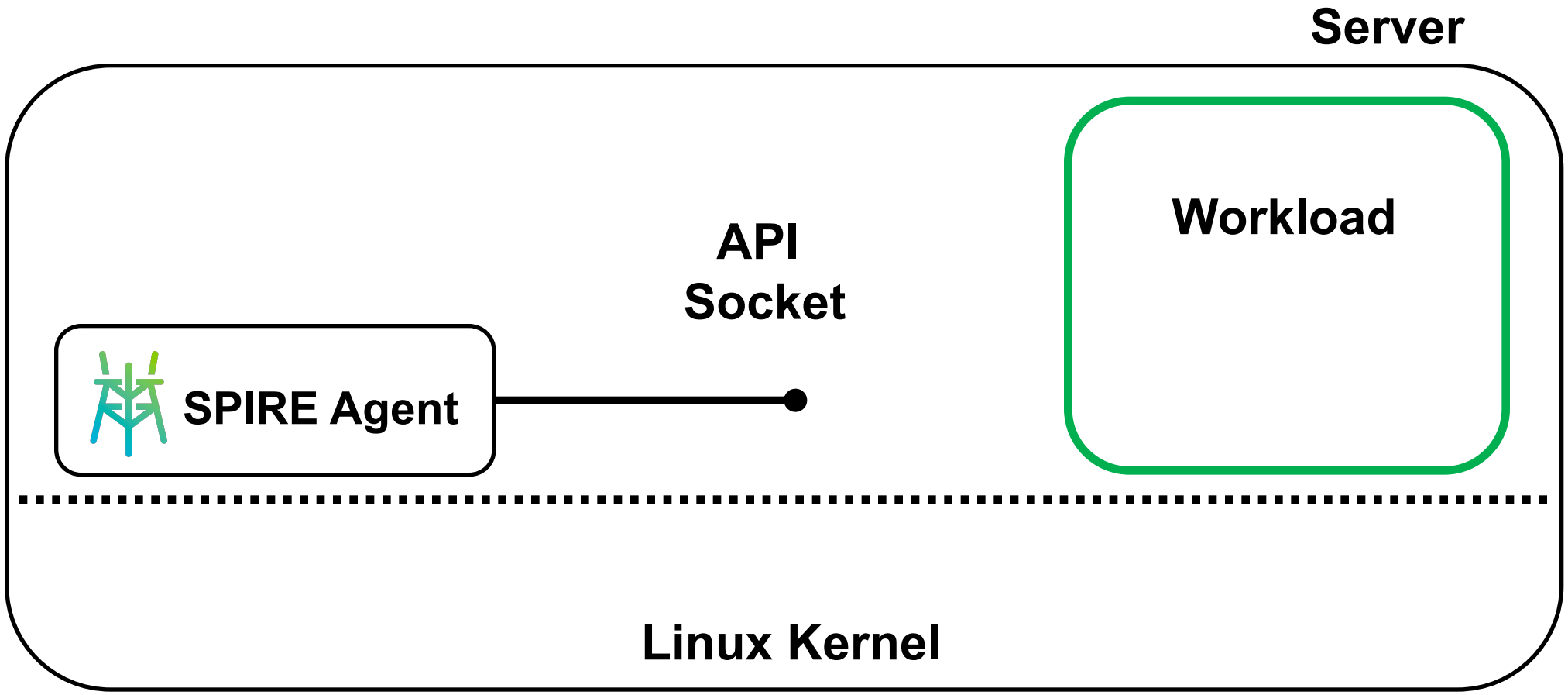
SPIRE Walkthrough



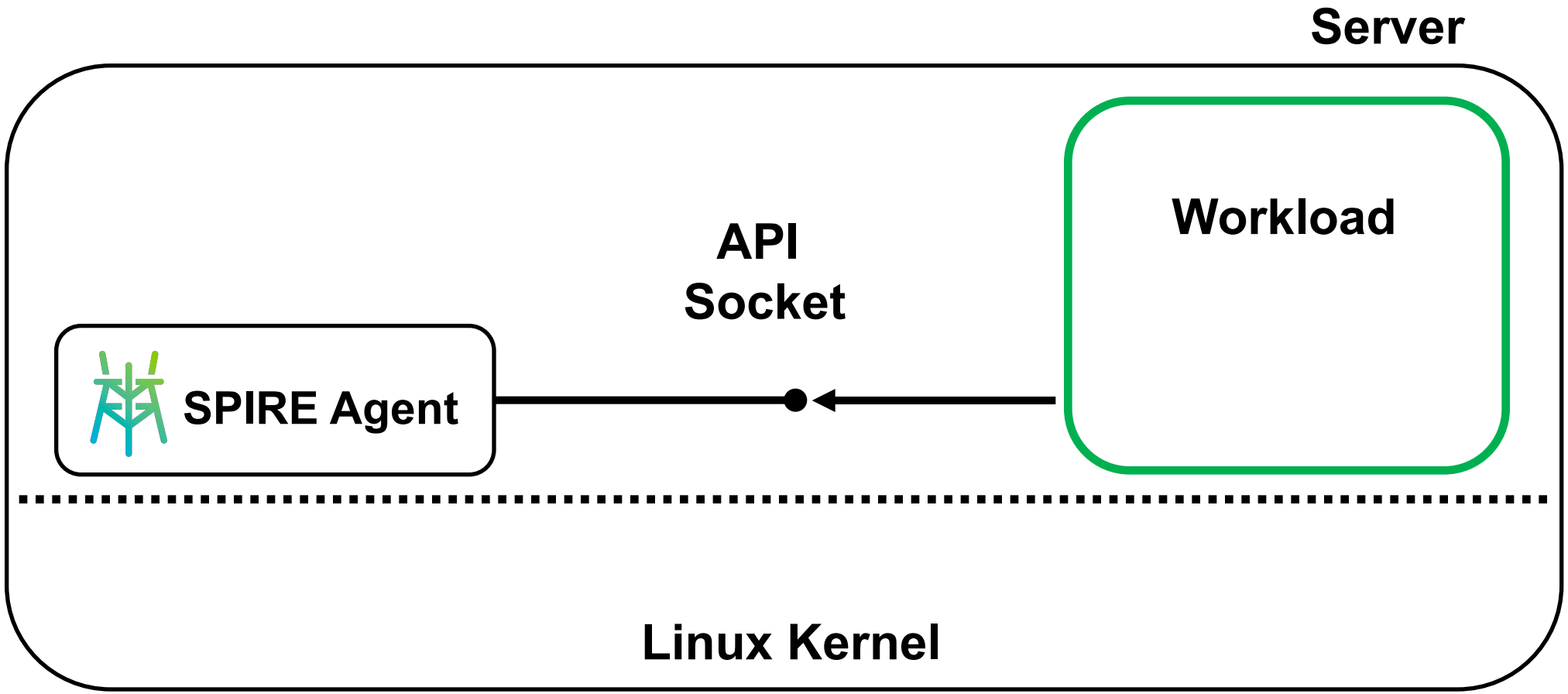
SPIRE Walkthrough



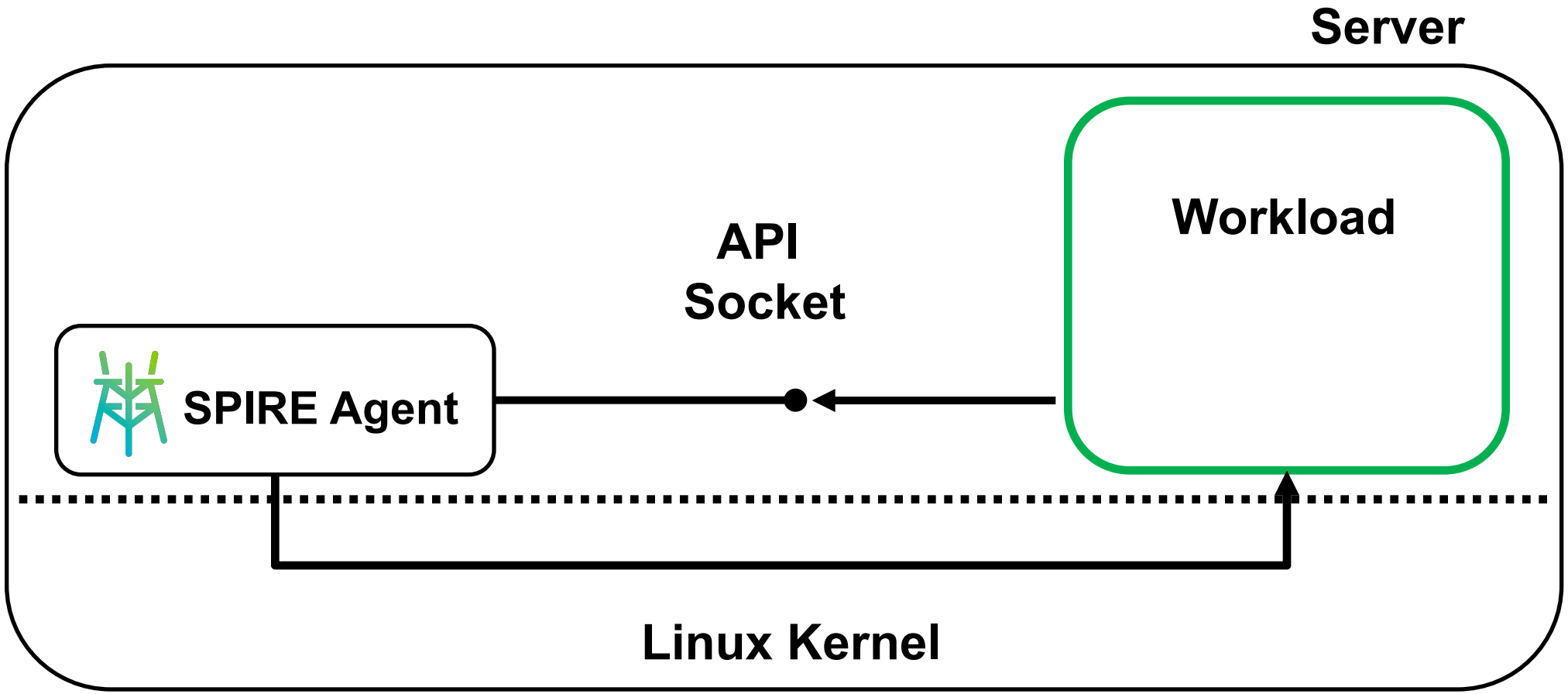
SPIRE Walkthrough



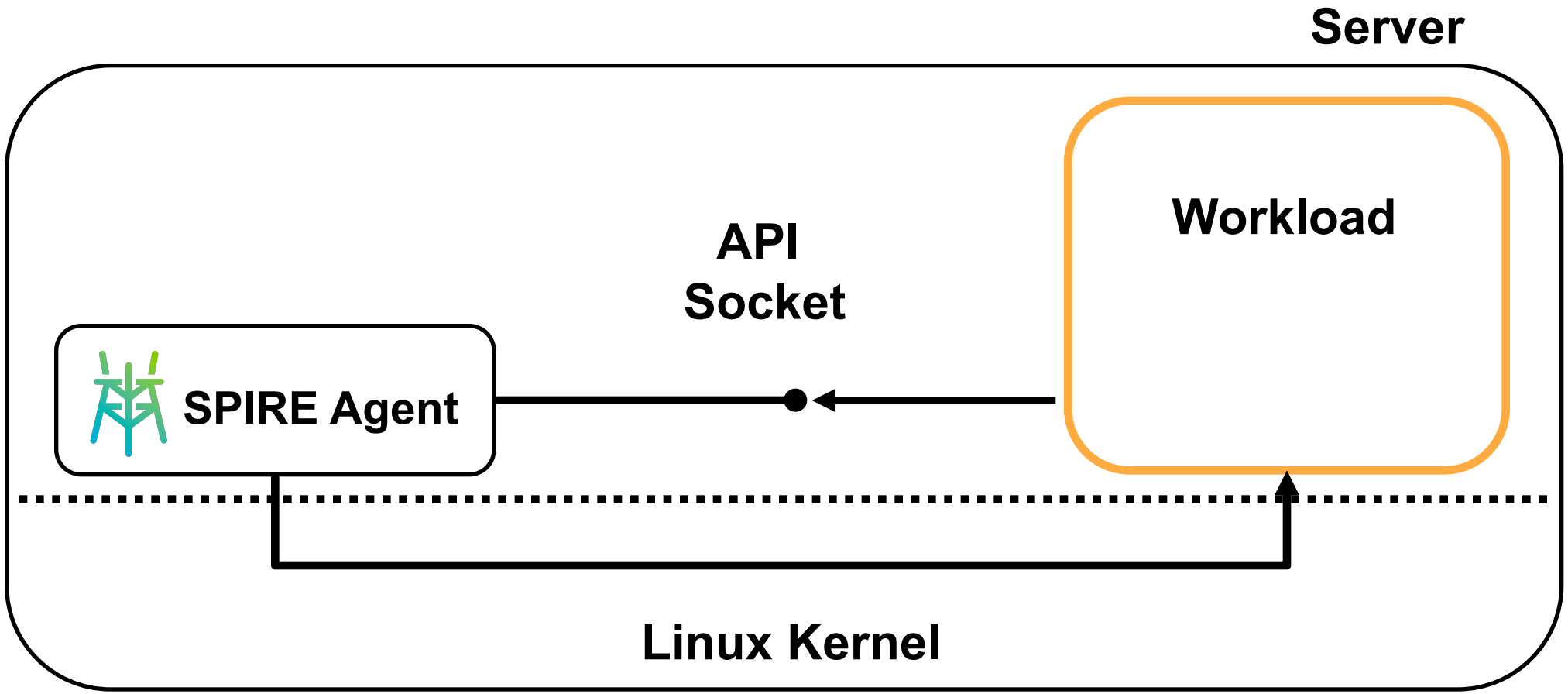
SPIRE Walkthrough



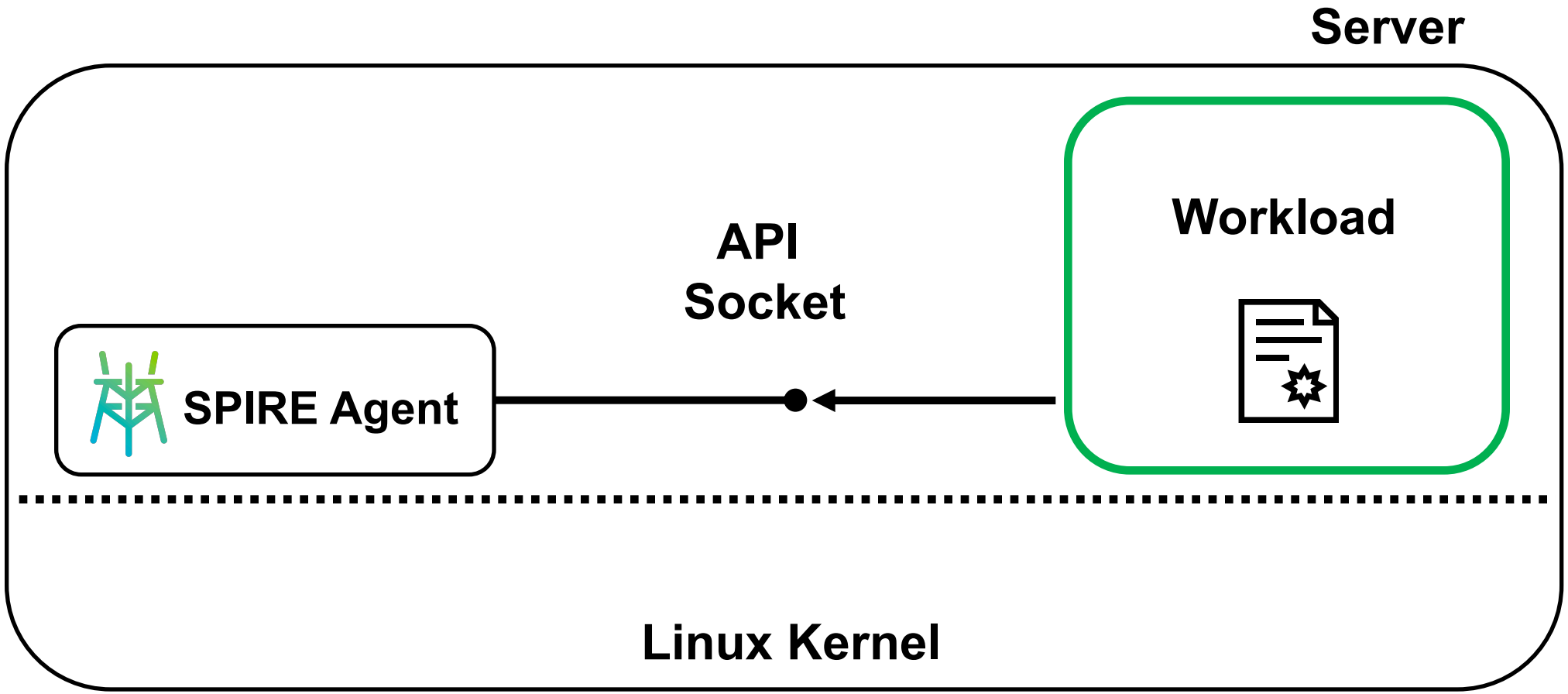
SPIRE Walkthrough



SPIRE Walkthrough



SPIRE Walkthrough



SPIRE Overview