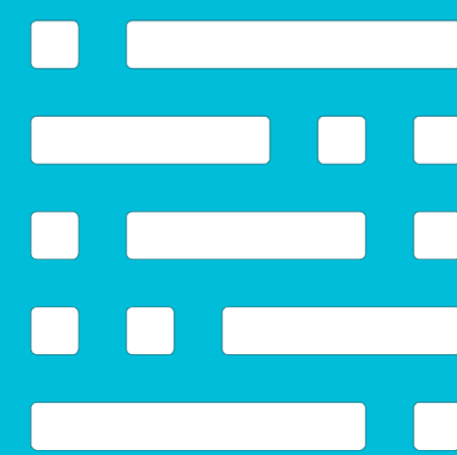


Scrutinizing SPIRE to Sensibly Strengthen SPIFFE Security

Evan Gilman, Matt Moyer



SPIRE



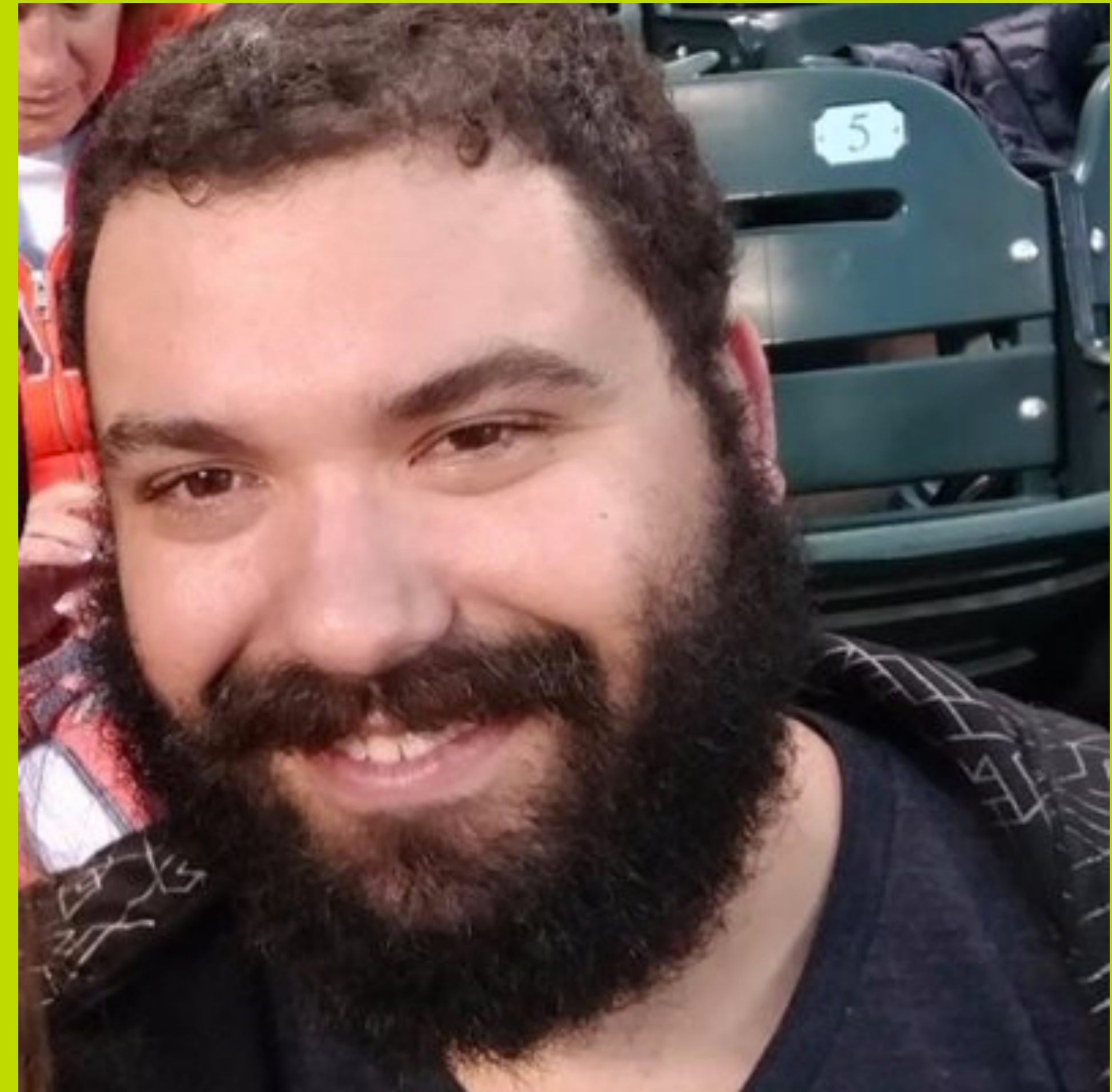
spiffe

 @moyerma

 @evan2645

Evan Gilman

Engineer at Scytale



 @moyerma
 @evan2645



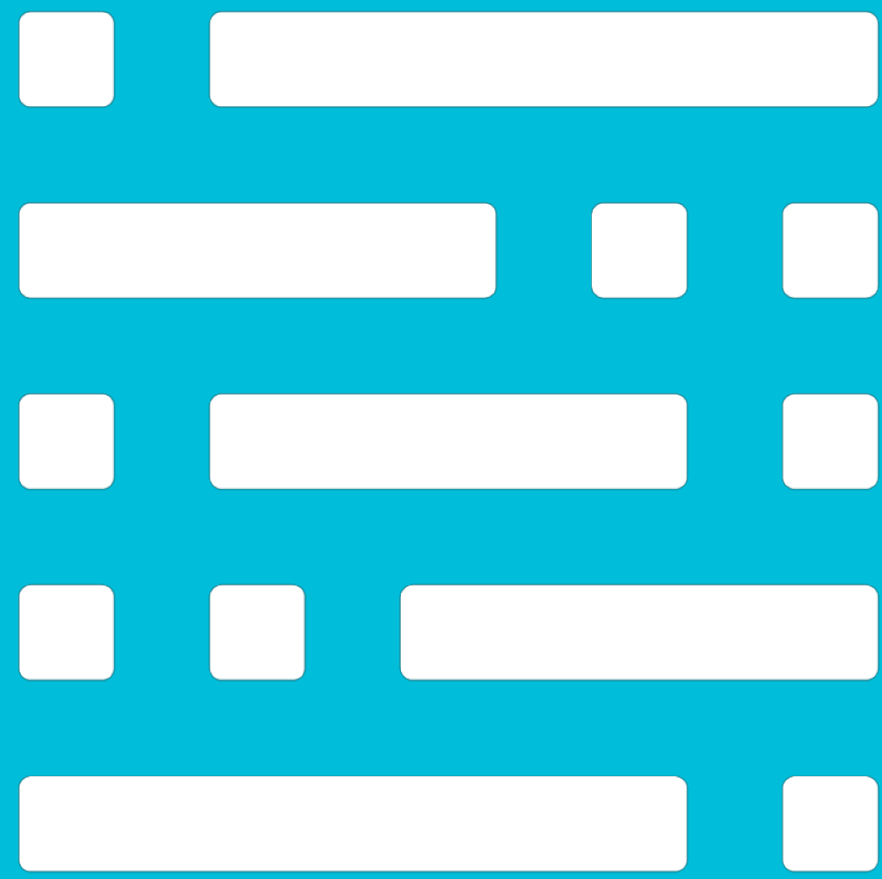
Matt Moyer

Engineer at Heptio



 @moyerma

 @evan2645



spiffie

What is SPIFFIE?

Secure Production Identity
Framework for Everyone



SPIFFE is a specification.



 @moyerma
 @evan2645

 SPIRE

SPIFFE ID

spiffe://example.org/foo

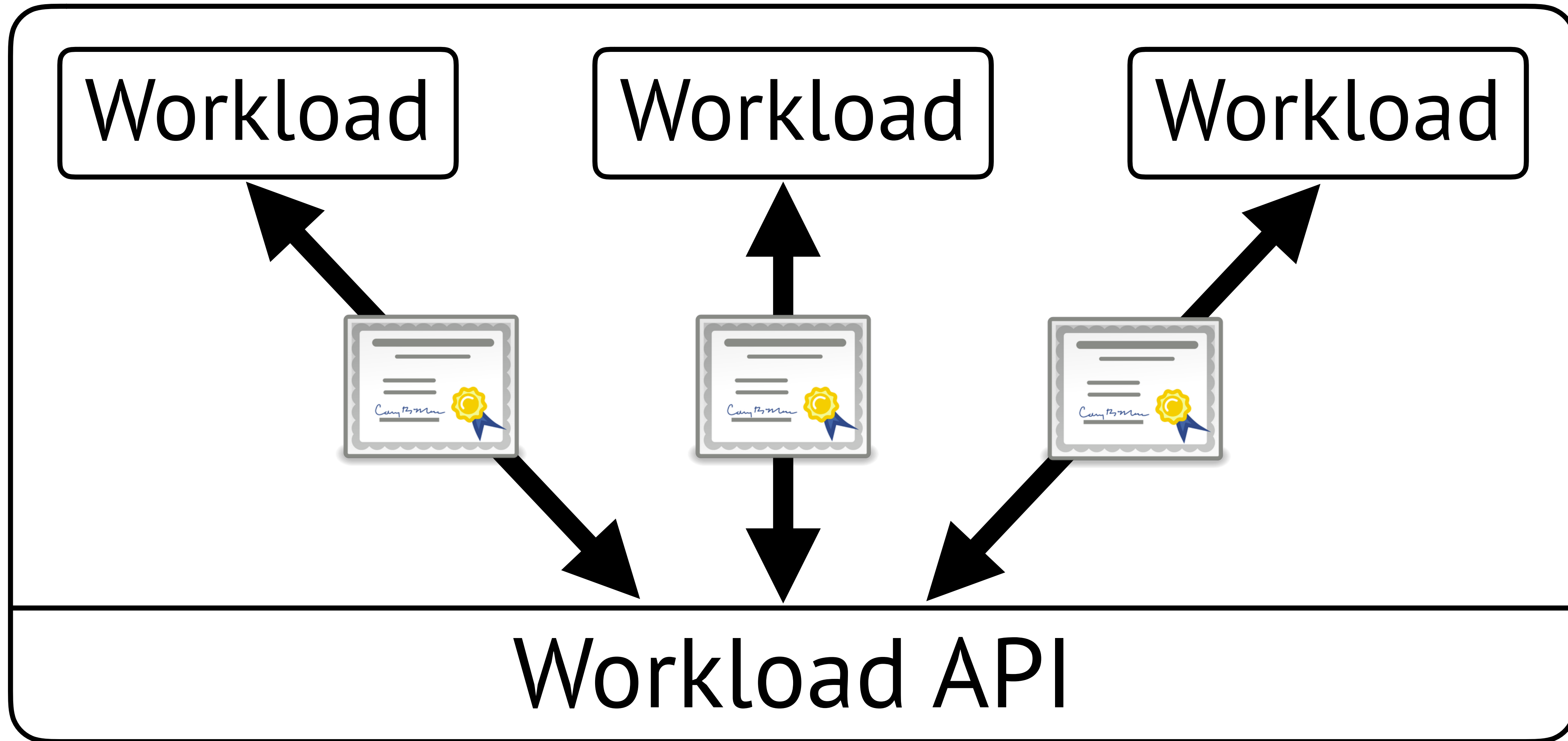
SPIFFE Verifiable Identity Document

spiffe://example.org/foo



SPIFFE Workload API

Server





What is SPIRE?

SPIFFE Runtime Environment



SPIRE is an (open source) implementation.



 @moyerma
 @evan2645

 SPIRE

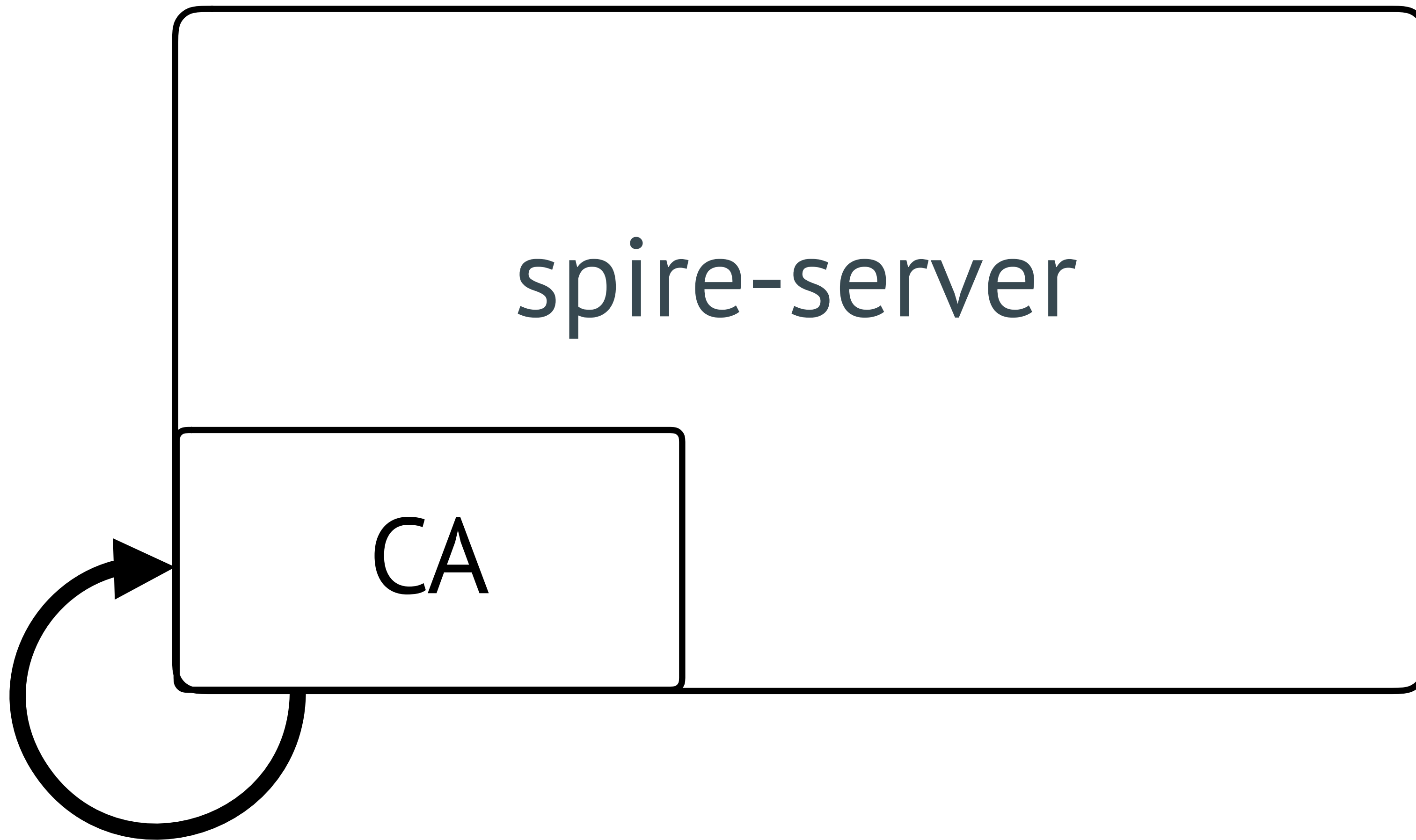
spire-server

- Identity Mapping
- Node Attestation
- SVID Issuance

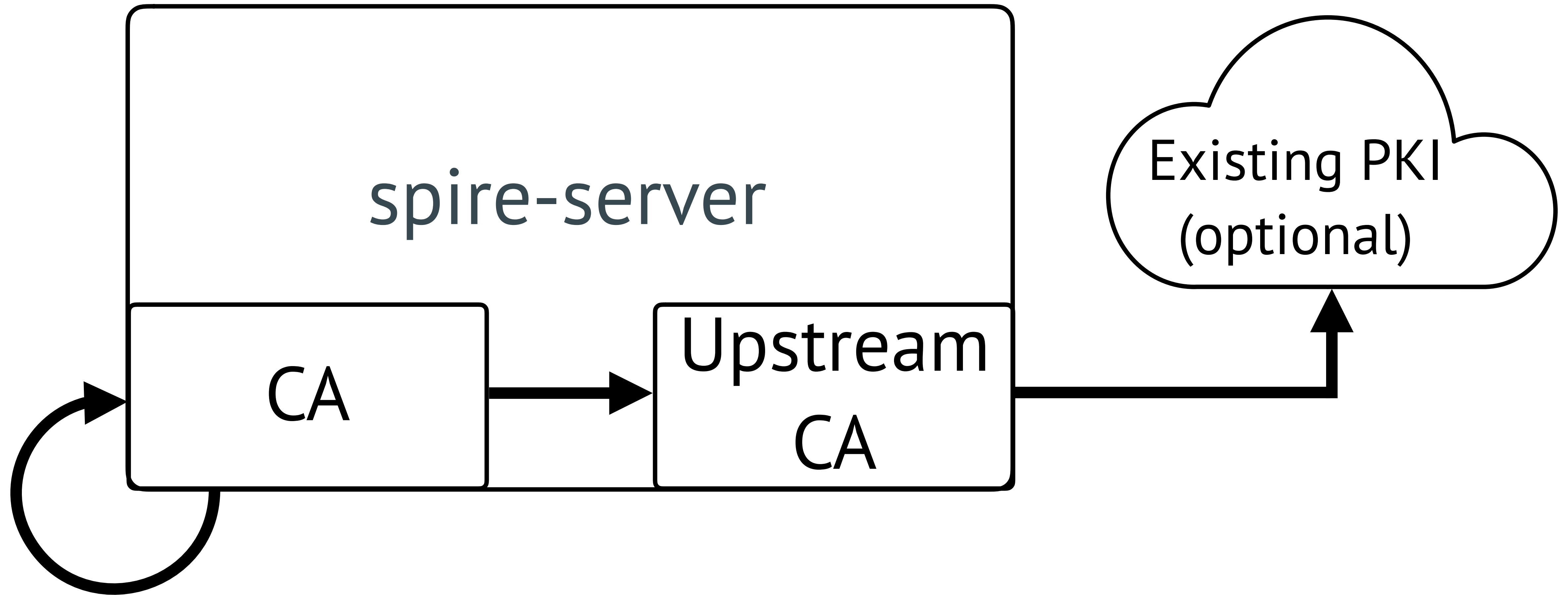
spire-agent

- Workload Attestation
- Workload API

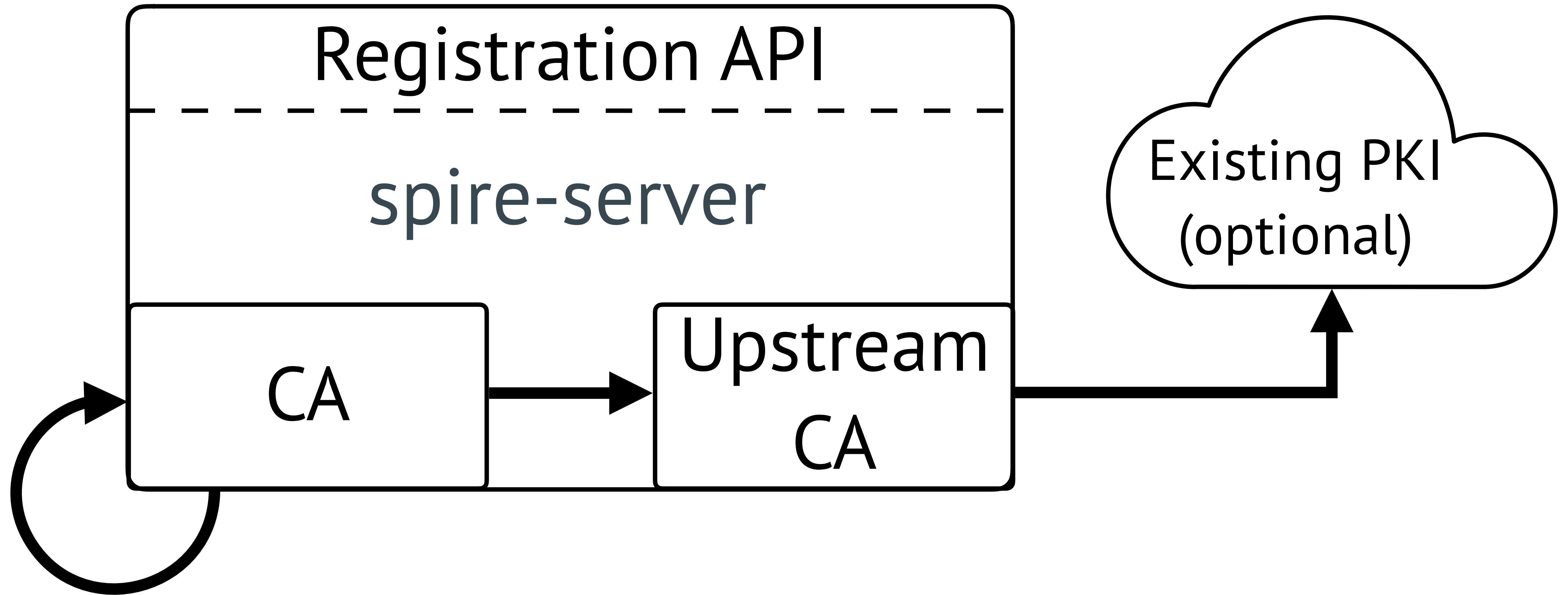
SPIRE Walkthrough



SPIRE Walkthrough



SPIRE Walkthrough



SPIRE Walkthrough

Parent ID: `spiffe://example.org/k8s/cluster/foo`

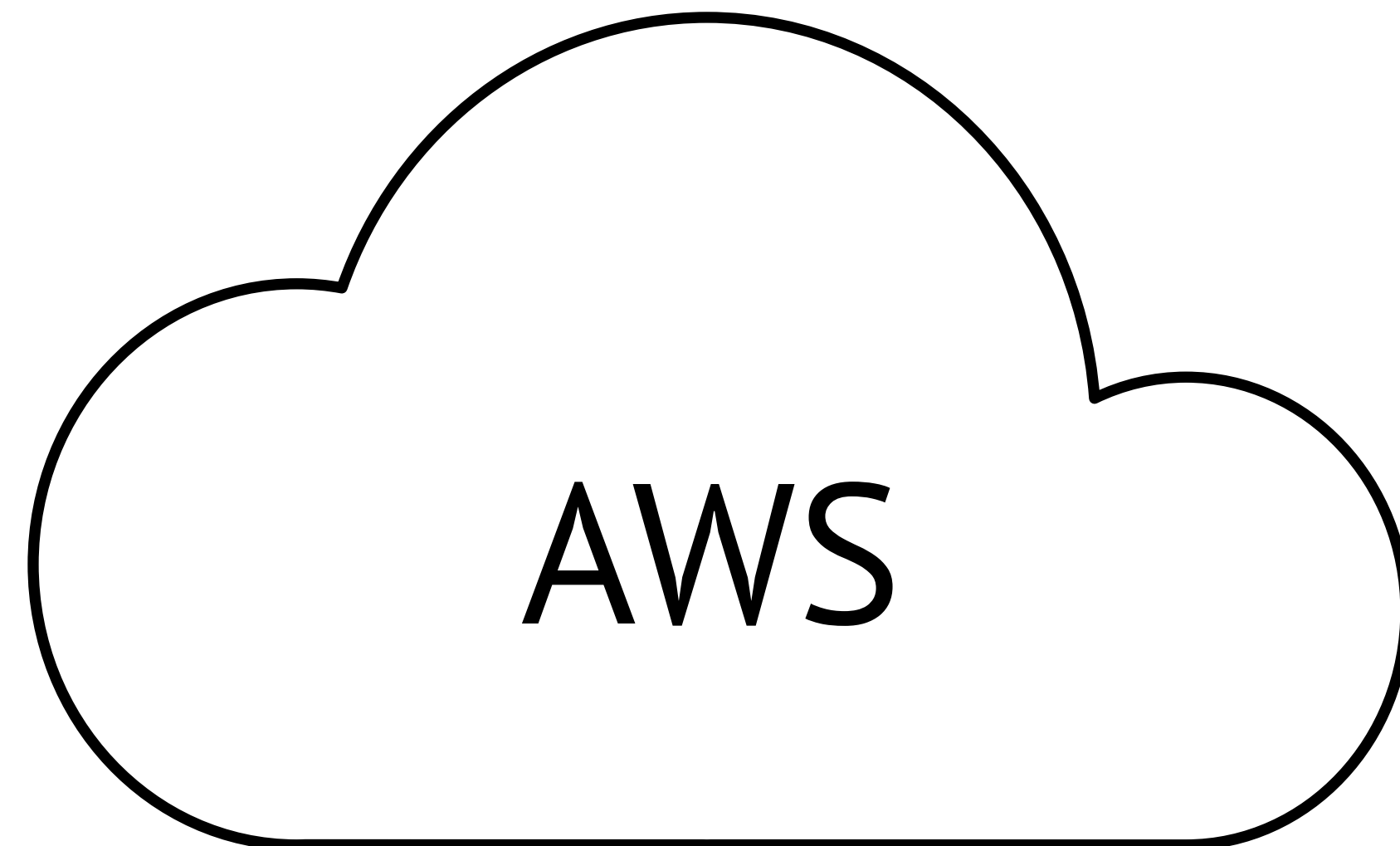
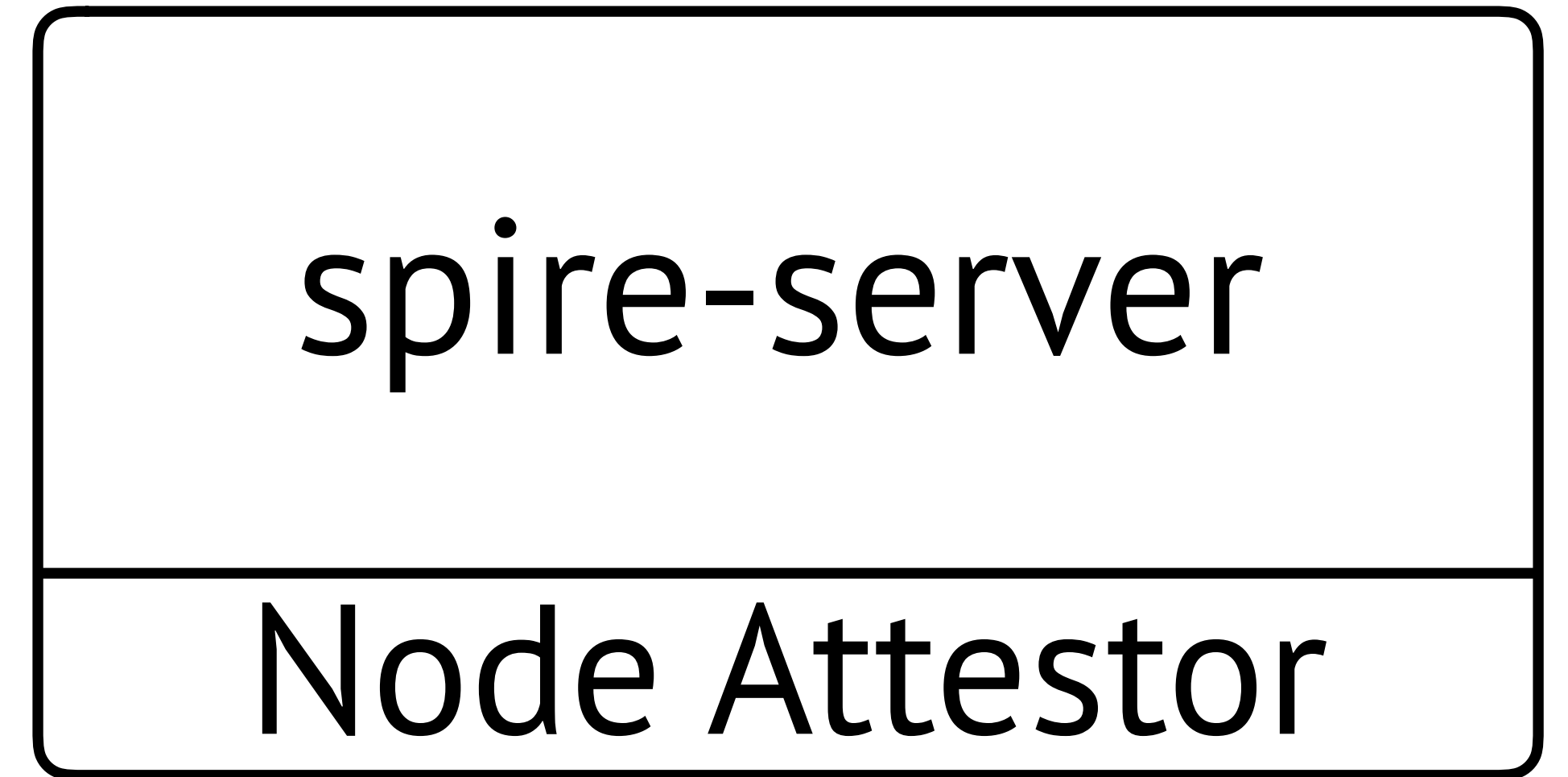
Selector: `k8s:ns:operations`

Selector: `k8s:sa:mediawiki`

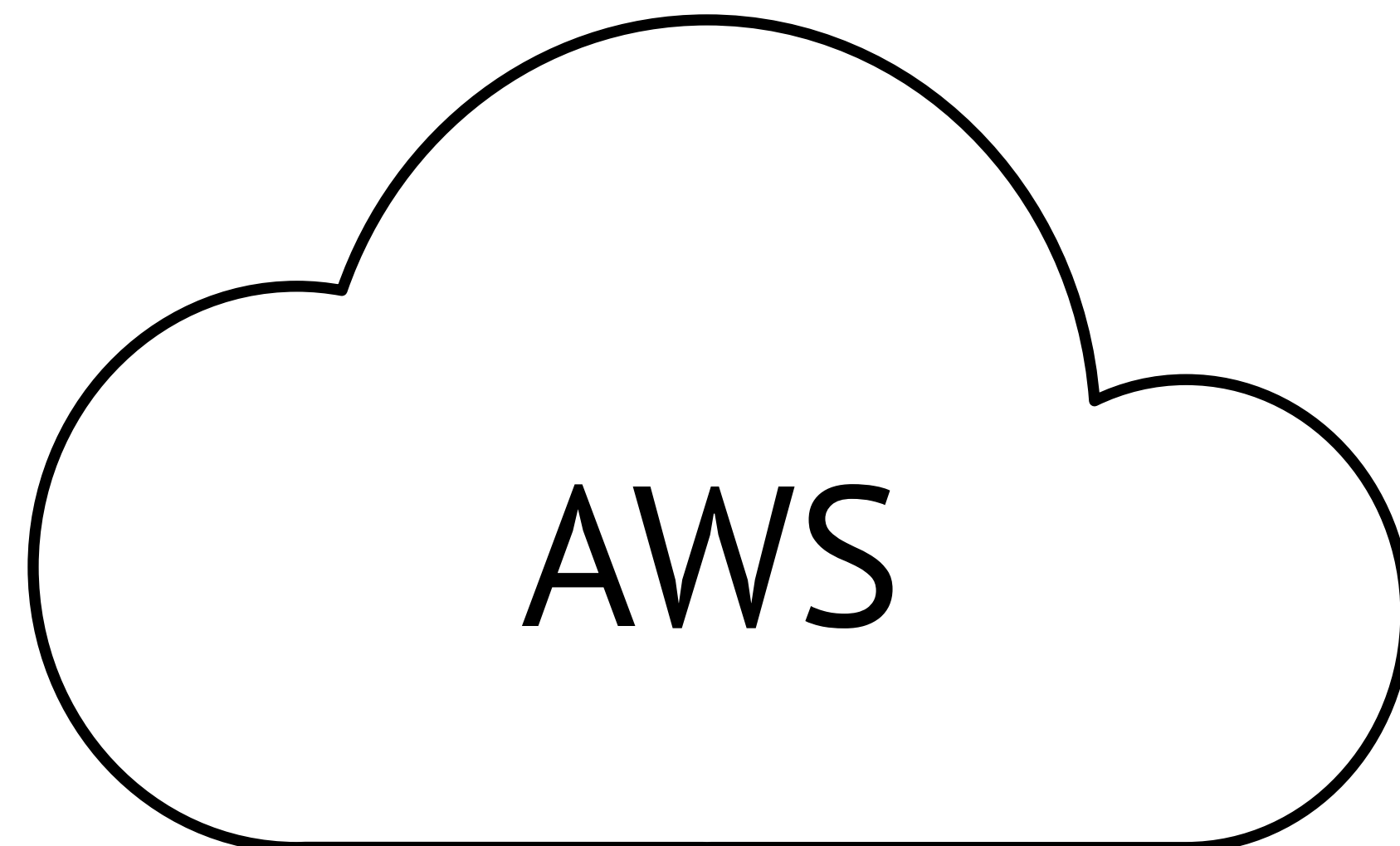
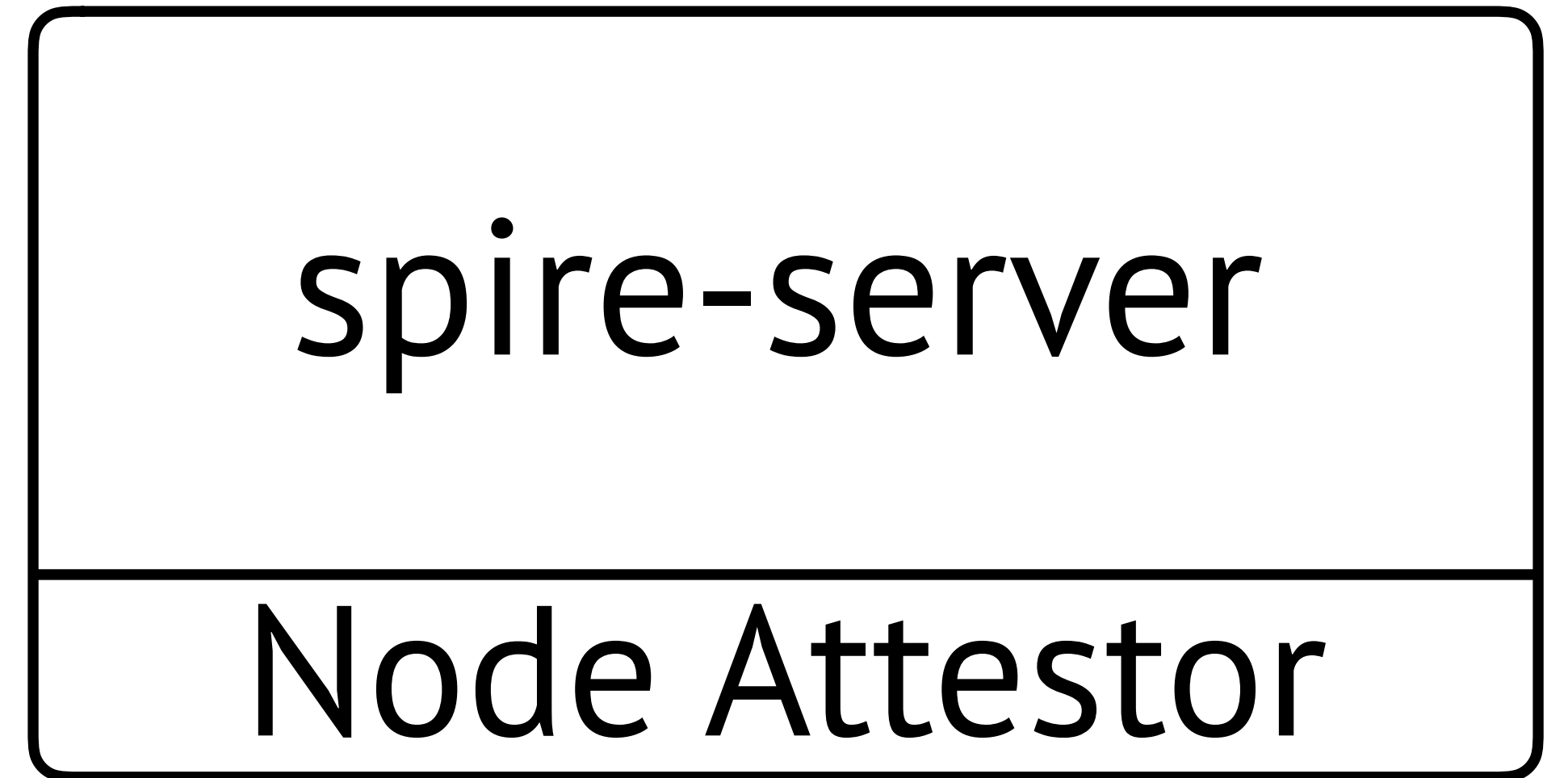
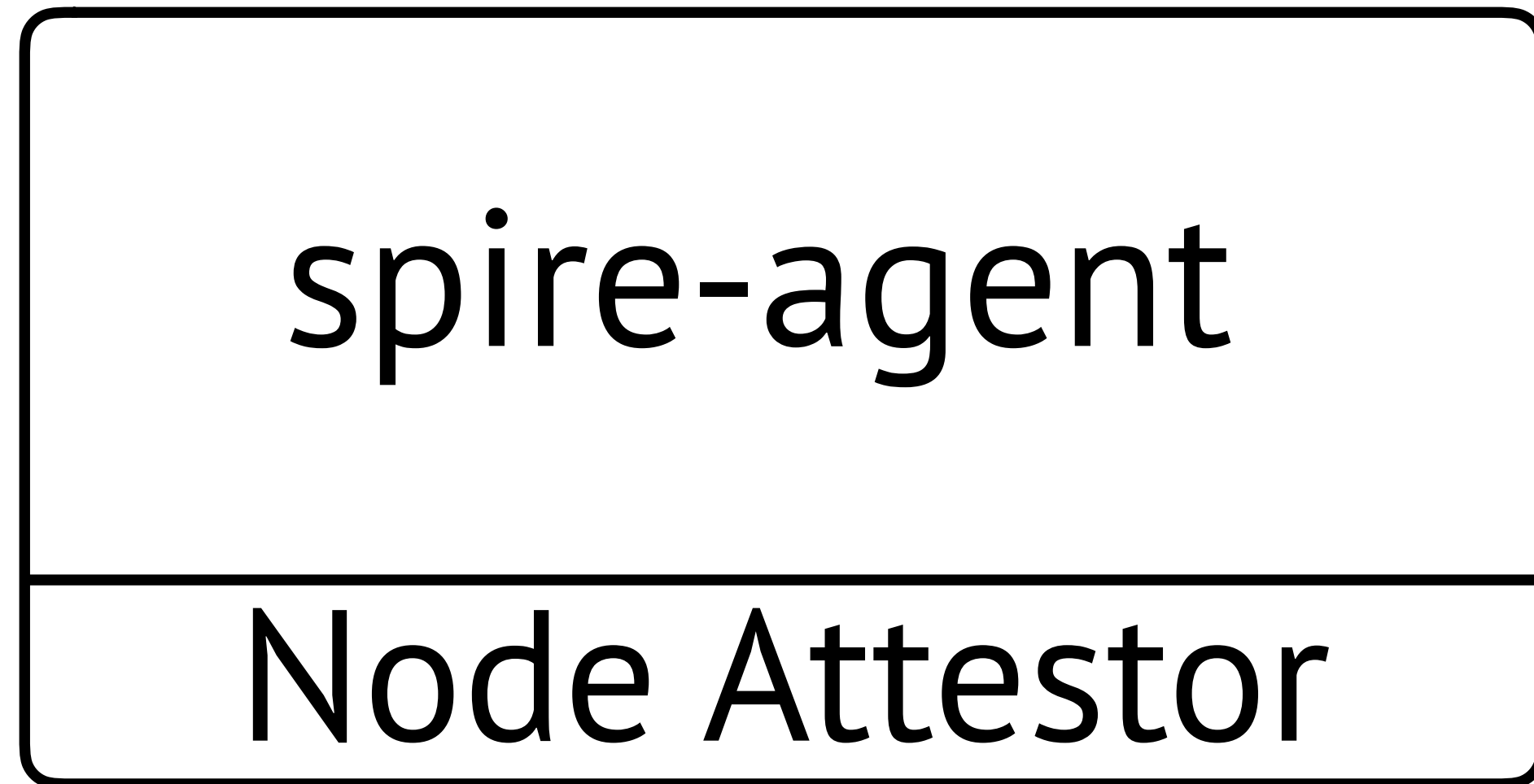
Selector: `docker:image-id:746b819f315e`

SPIFFE ID: `spiffe://example.org/ops/wiki`

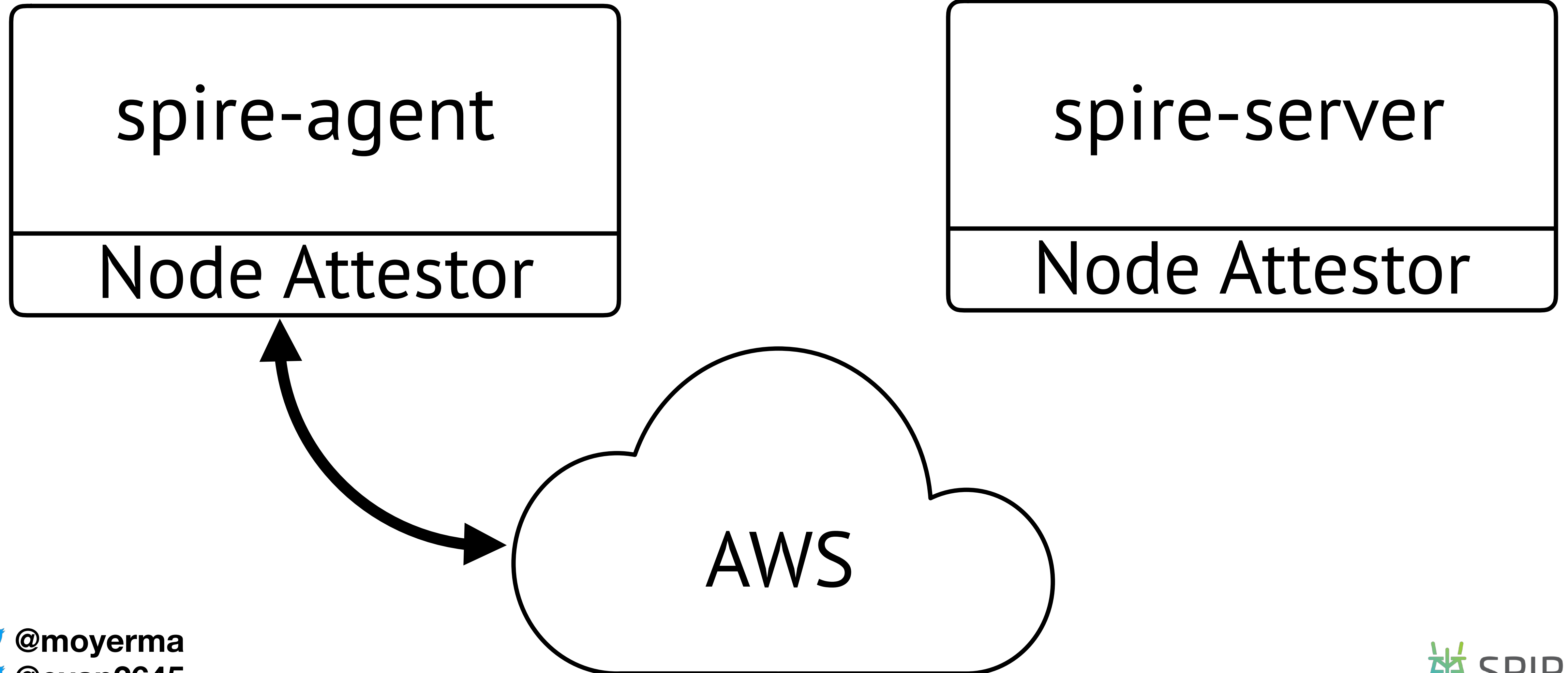
SPIRE Walkthrough



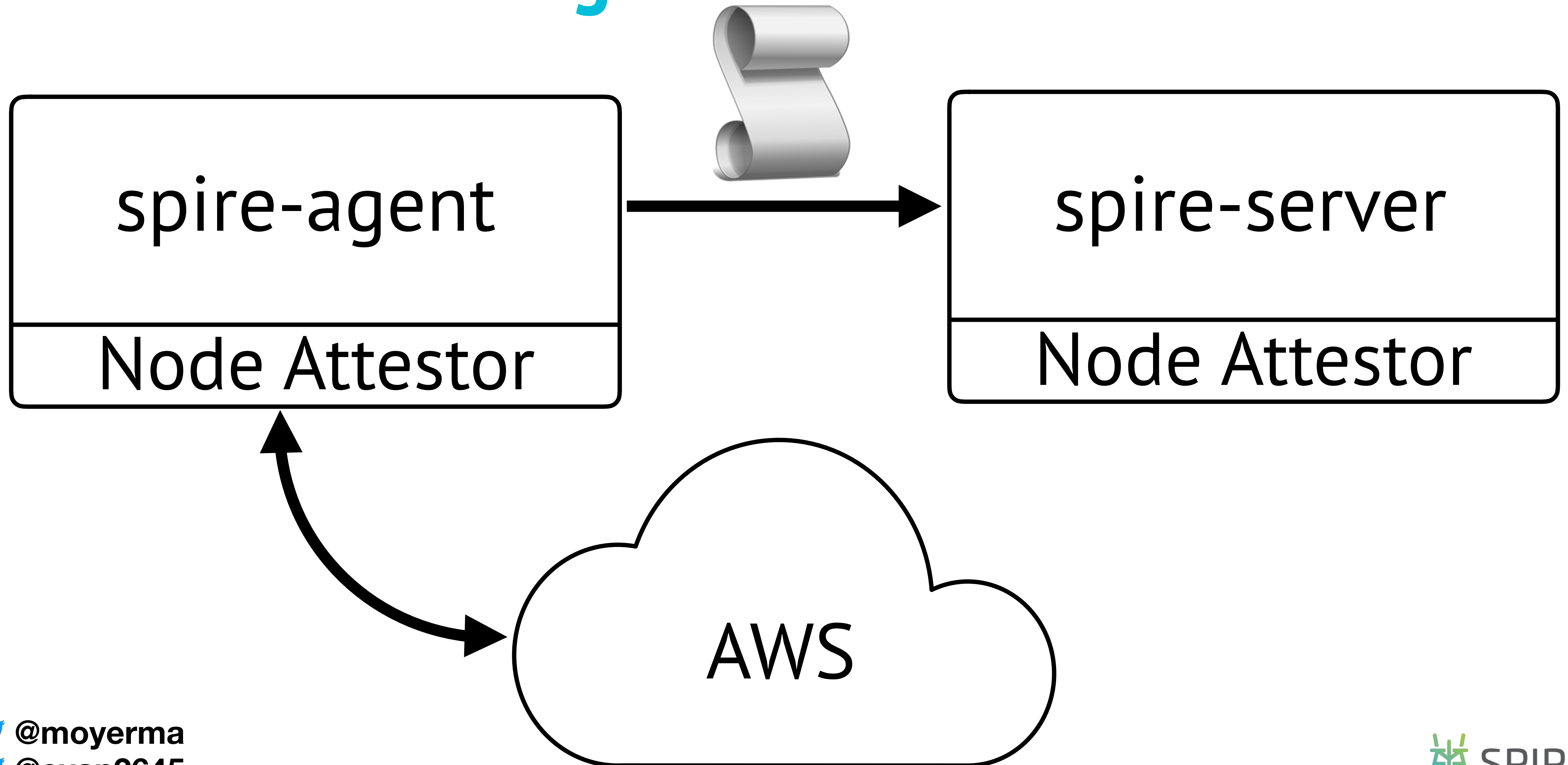
SPIRE Walkthrough



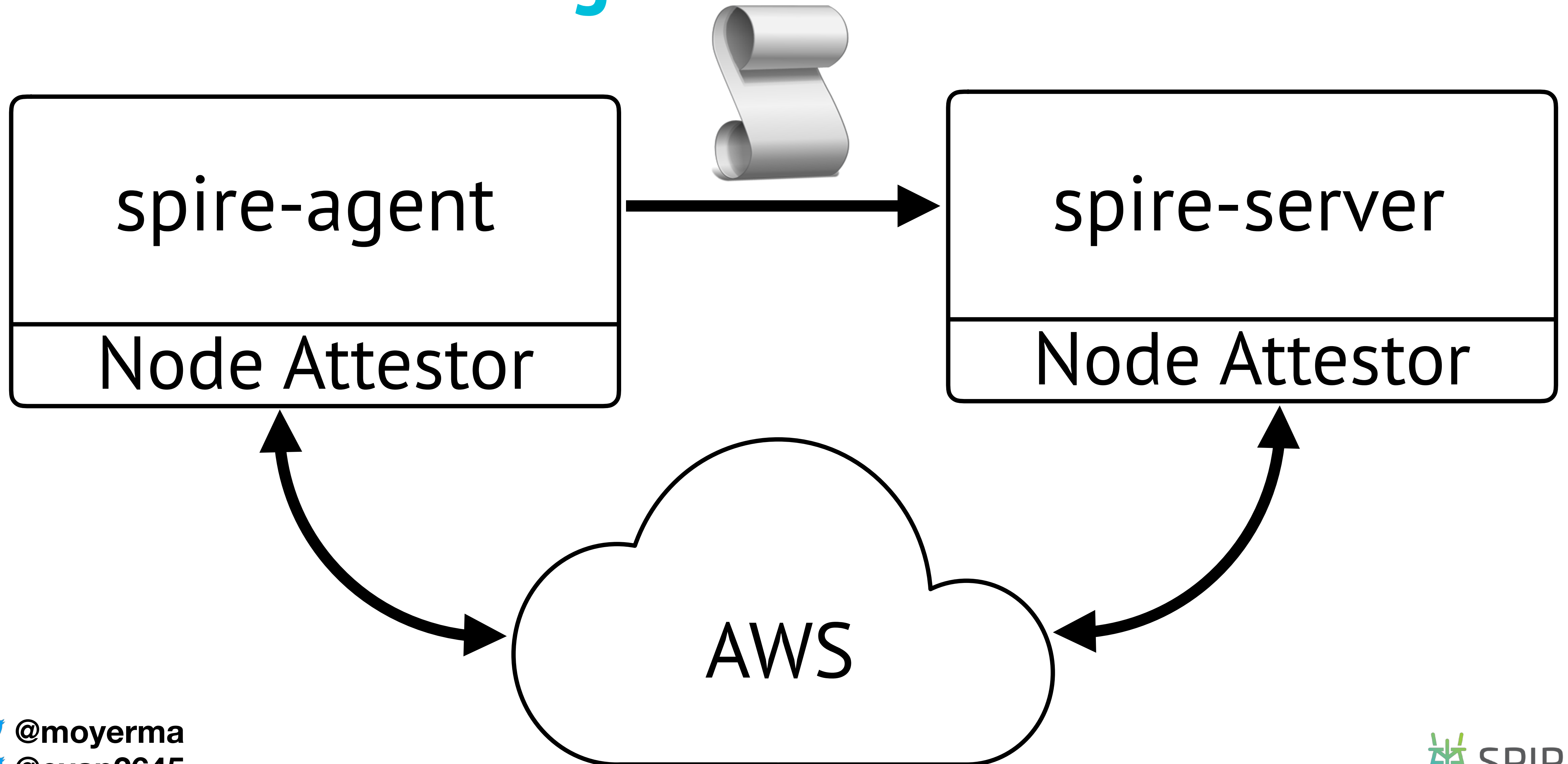
SPIRE Walkthrough



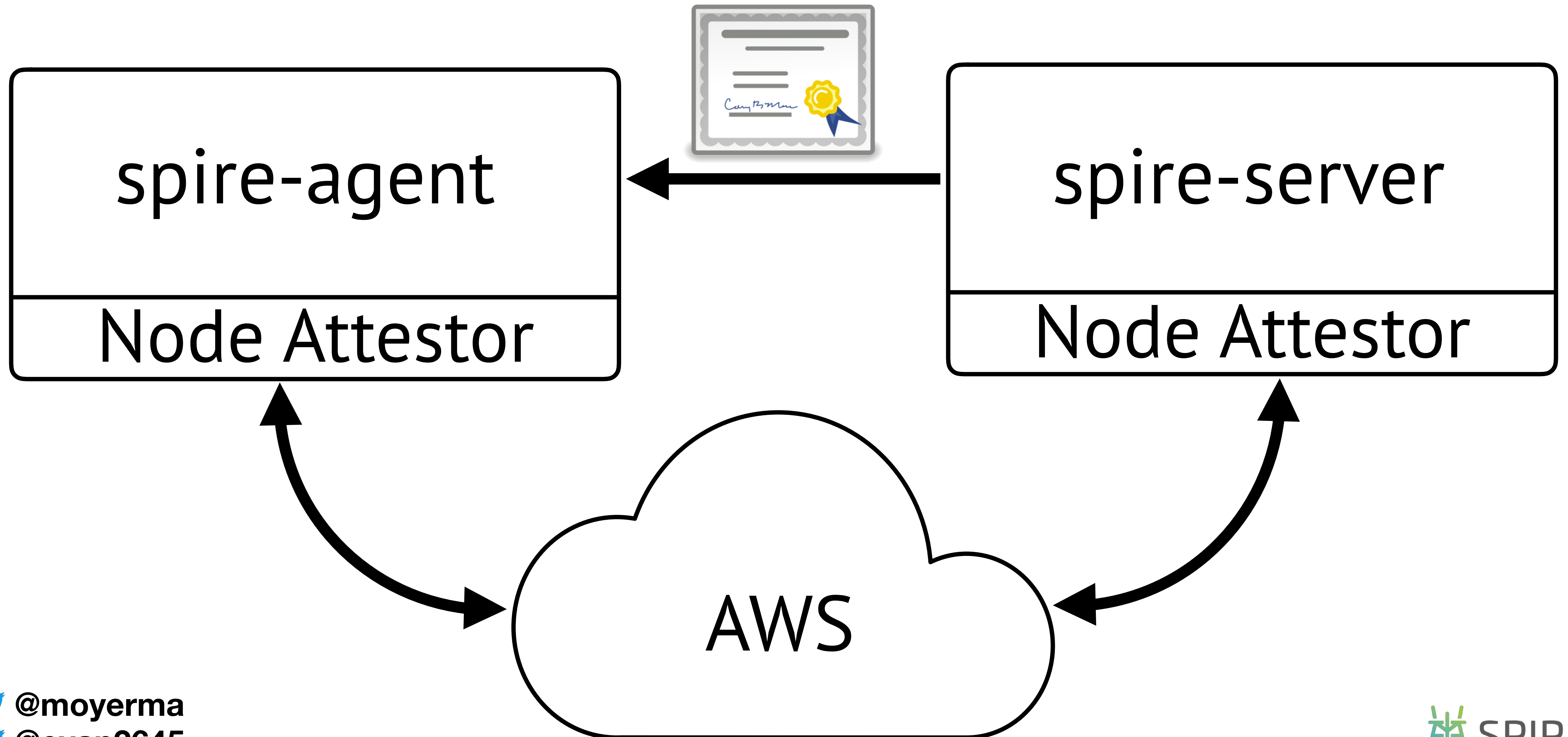
SPIRE Walkthrough



SPIRE Walkthrough

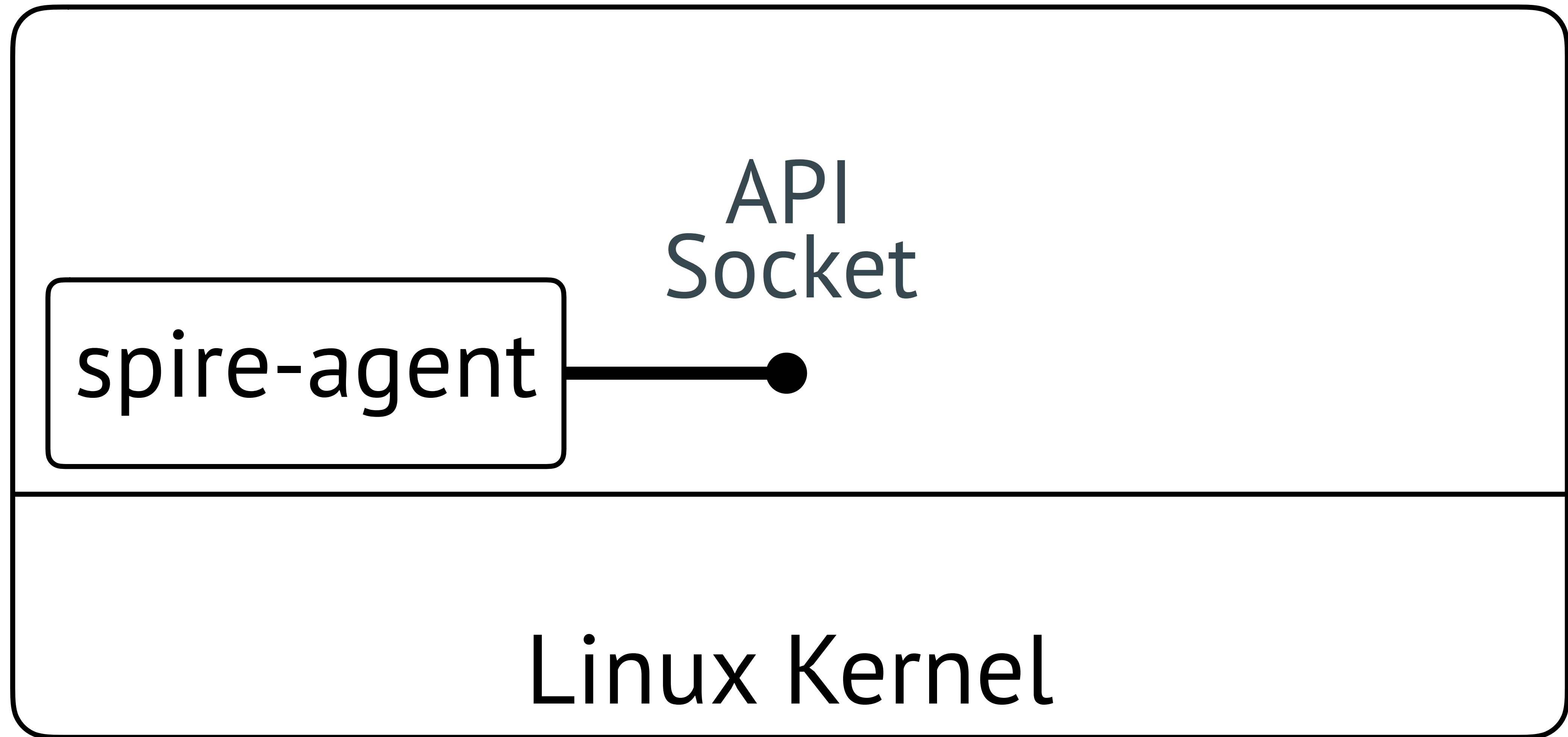


SPIRE Walkthrough



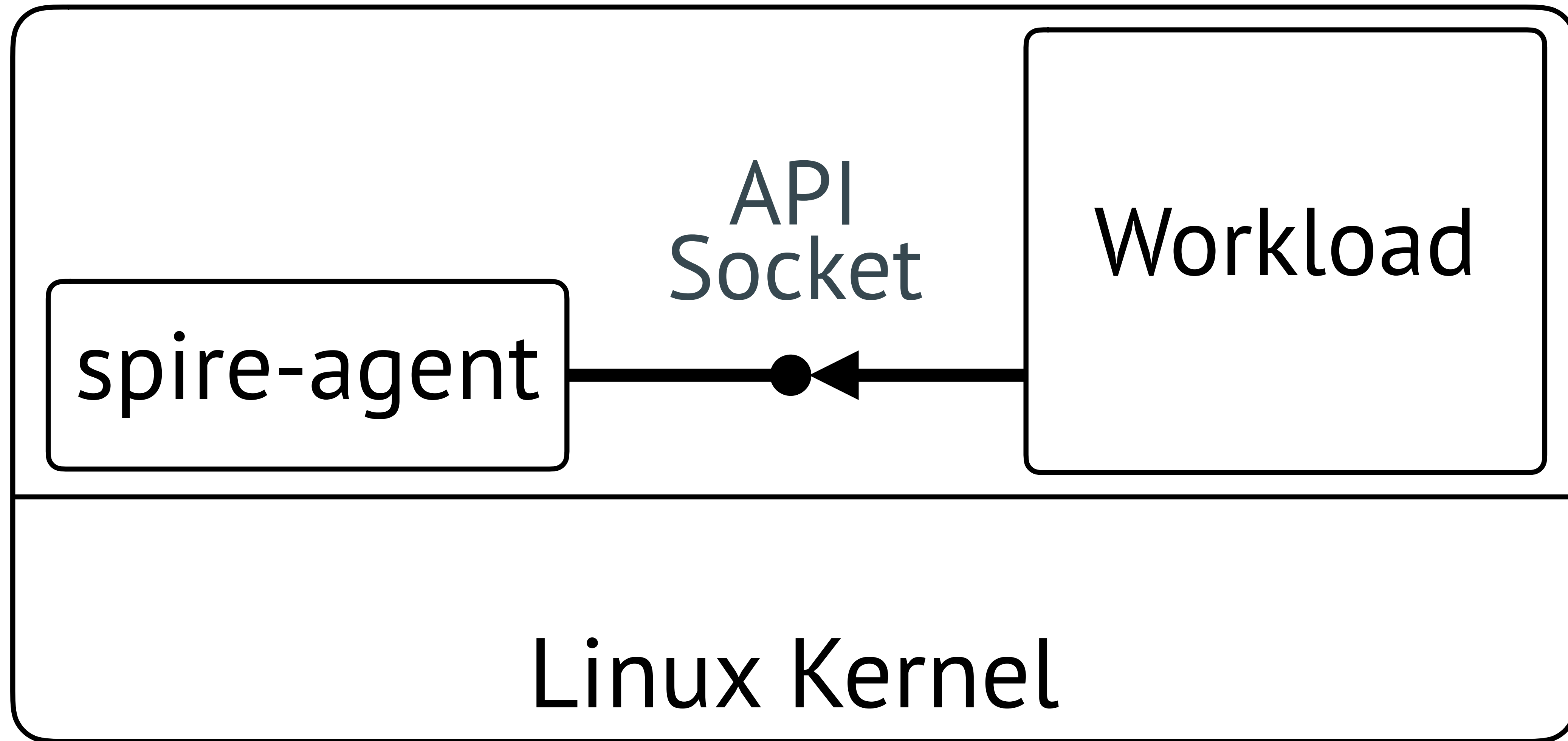
SPIRE Walkthrough

Server



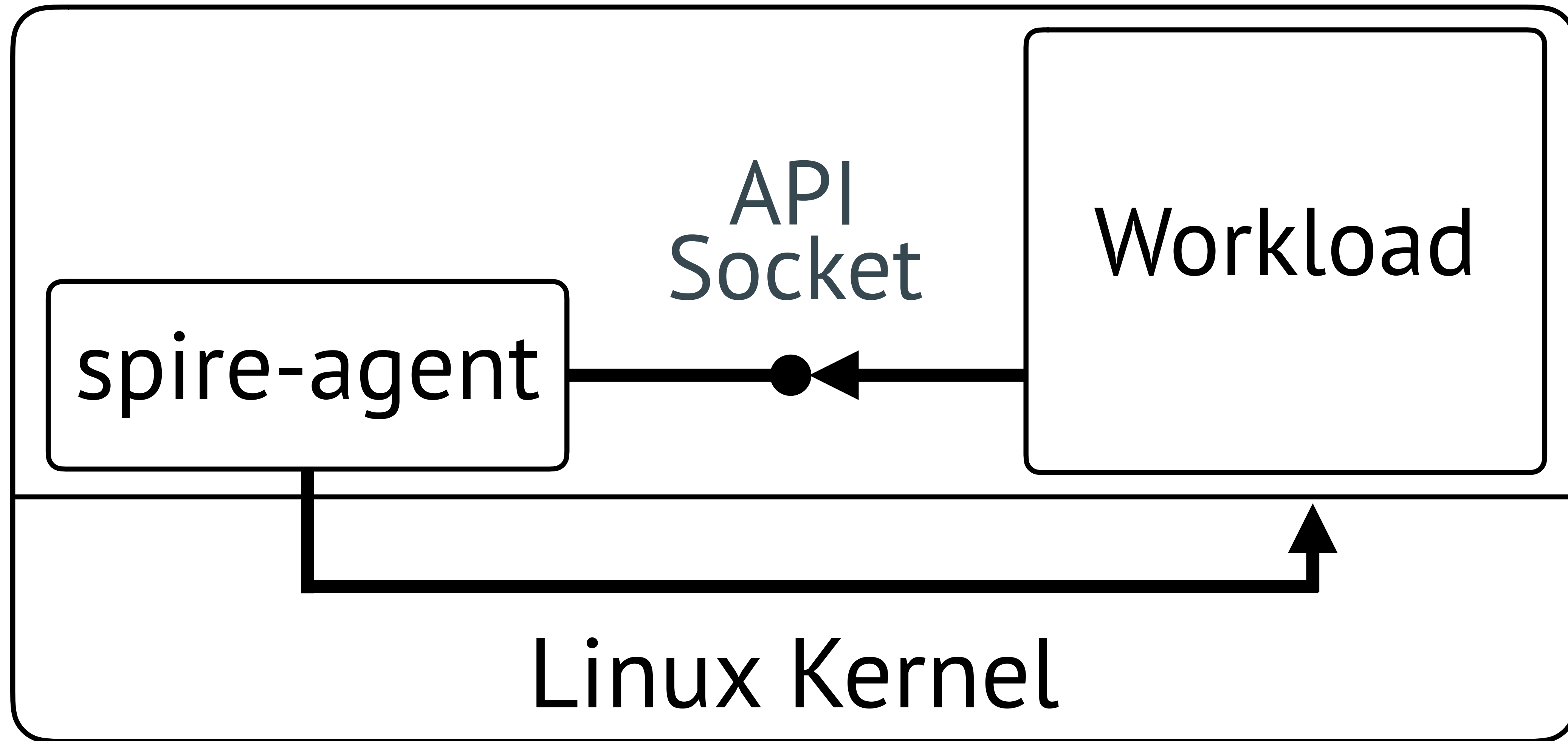
SPIRE Walkthrough

Server



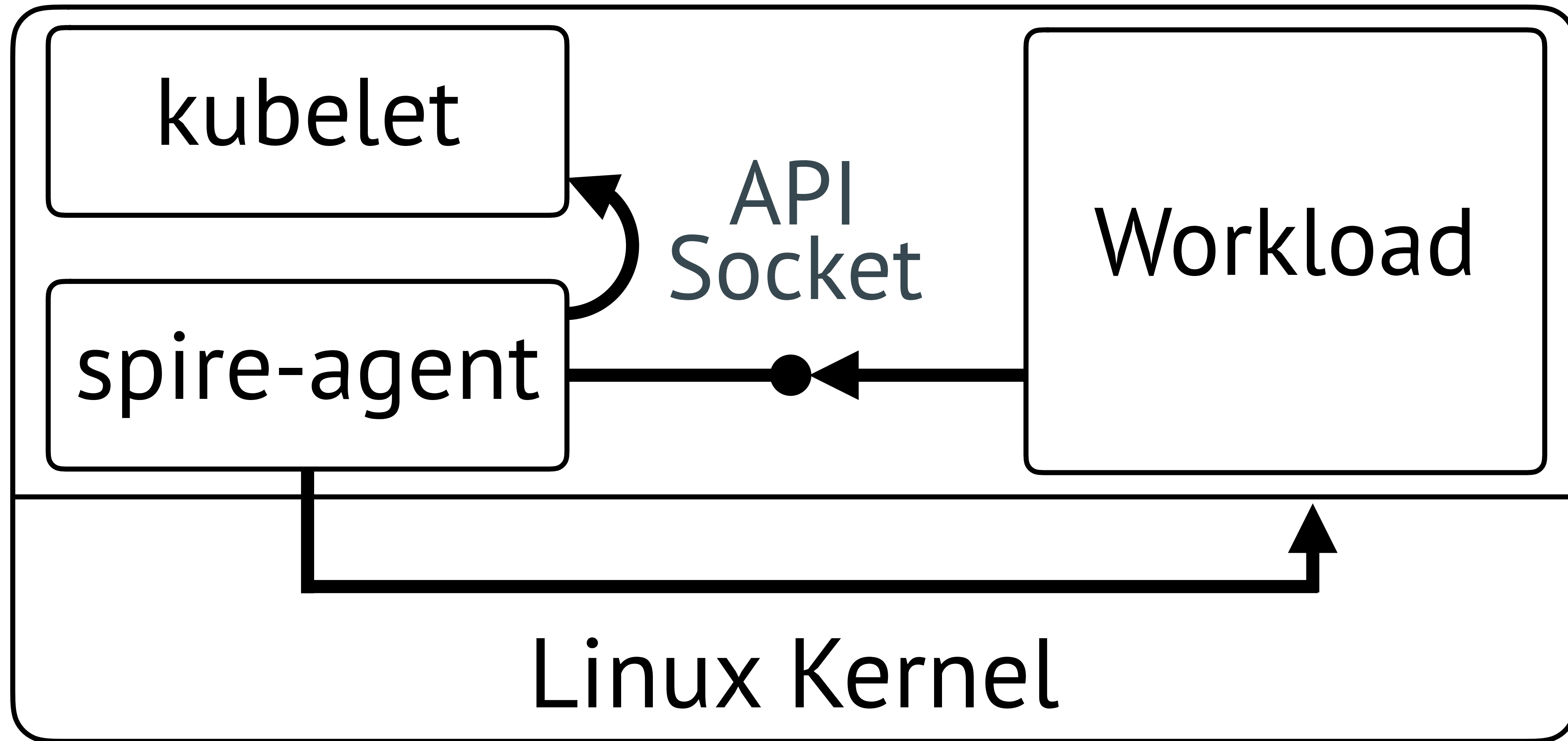
SPIRE Walkthrough

Server



SPIRE Walkthrough

Server





What is this all about?

 @moyerma
 @evan2645

 SPIRE

Community Security Modeling Exercise

- Study SPIRE and describe its security model.
- Kicked off December 2017 in collaboration with:



Justin Cappos (NYU)



Enrico Schiattarella

Goals

1. Understand the expected security properties of a SPIFFE implementation.
2. Explore what vulnerability classes exist in a SPIFFE system.
3. Identify how SPIRE (the implementation) can be improved.
4. Identify how SPIFFE (the specification) can be improved.

Non-Goals

1. Identify specific implementation vulnerabilities in SPIRE.
2. Formally prove anything about SPIFFE/SPIRE.

A black and white photograph of a metal padlock on a chain, symbolizing security. The padlock is attached to a heavy metal chain that is looped around a vertical metal post. The background is blurred, showing what appears to be a fence or gate structure. The text is overlaid in the center of the image.

**What security properties
do we expect?**

Attacker Goals

- Impersonate a workload, node agent, or server.
- Deny service to workloads or node agents.
- Create a new (forged) identity.

Attacker Goals (cont.)

- Modify permissions associated with an identity.
- Trick a party into using the wrong identity.
- Compromise the software running on a system.

Attacker Starting Position

spire-server

spire-agent

workload

Attacker Superpowers

- Container escape vulnerability.
- Network-level interception/manipulation capability.
- Malformed certificate or CSR (short of RCE).
- Ability to overwhelm server with requests.

Attacker Superpowers (cont.)

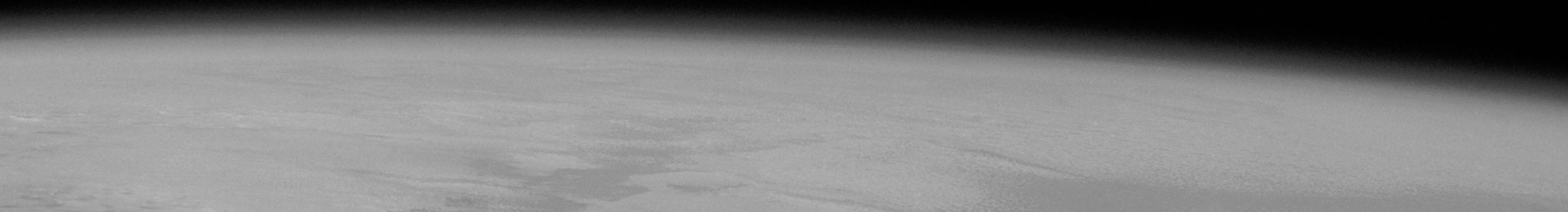
- Remote code execution reachable from...
 - Go CSR parser.
 - Go X.509 certificate parser.
 - Pre-authenticated protocol stack (TLS, HTTP, gRPC)

Attacker Superpowers (cont.)

- Exploit for explicitly mitigated problem.



(State) Space Explorers



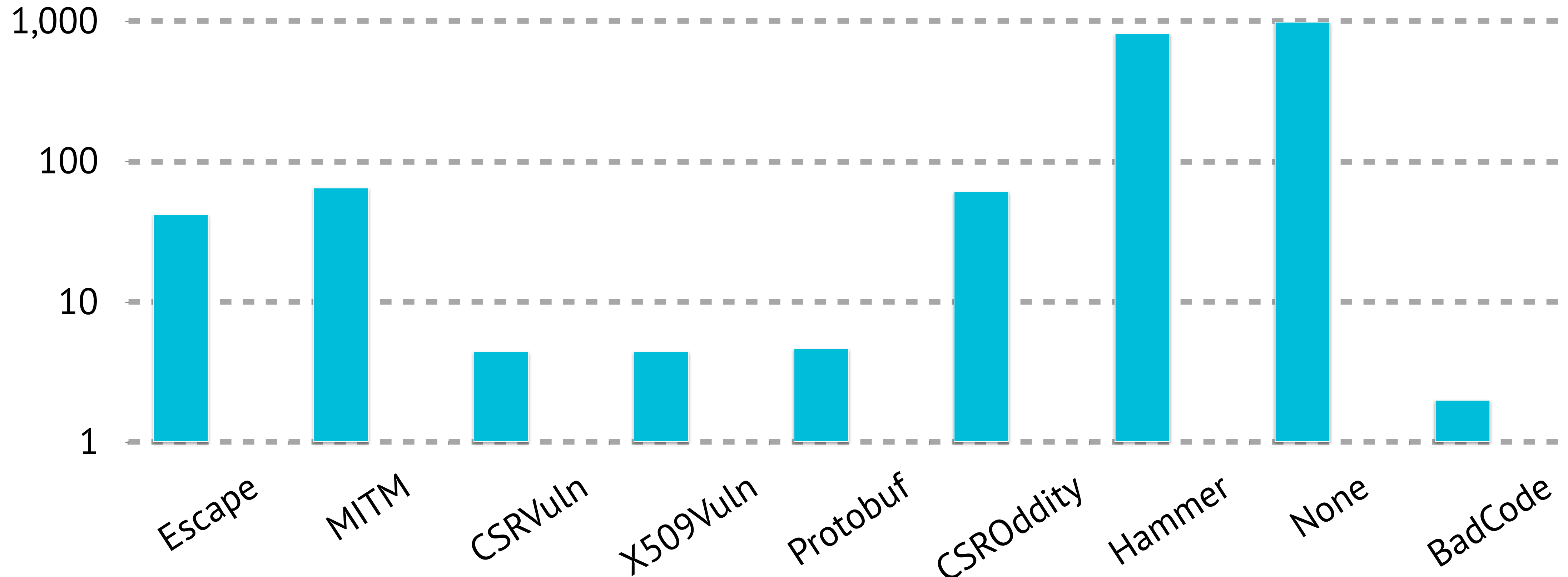
The Matrix

- We explored combinations of:
 - Attacker Goal
 - Starting Position
 - Victim Component
 - Attacker Capabilities
- This involved a lot of *talking*.
- The results went into a *big spreadsheet*.

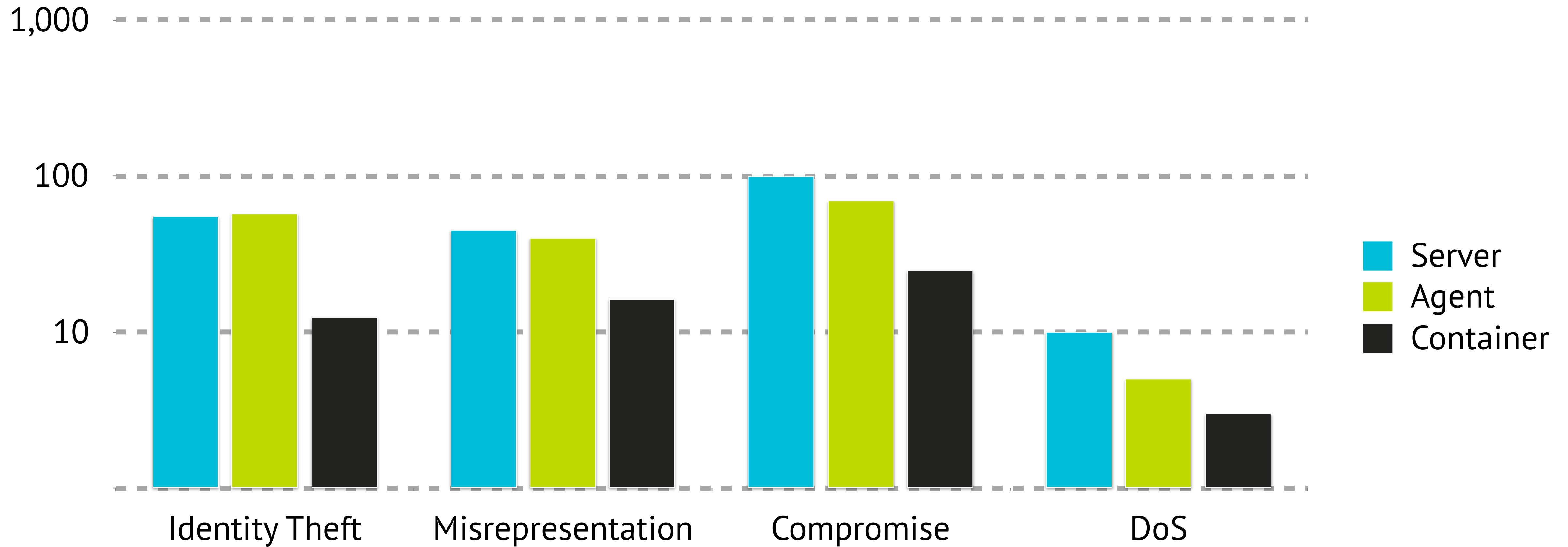
 @moyerma

 @evan2645

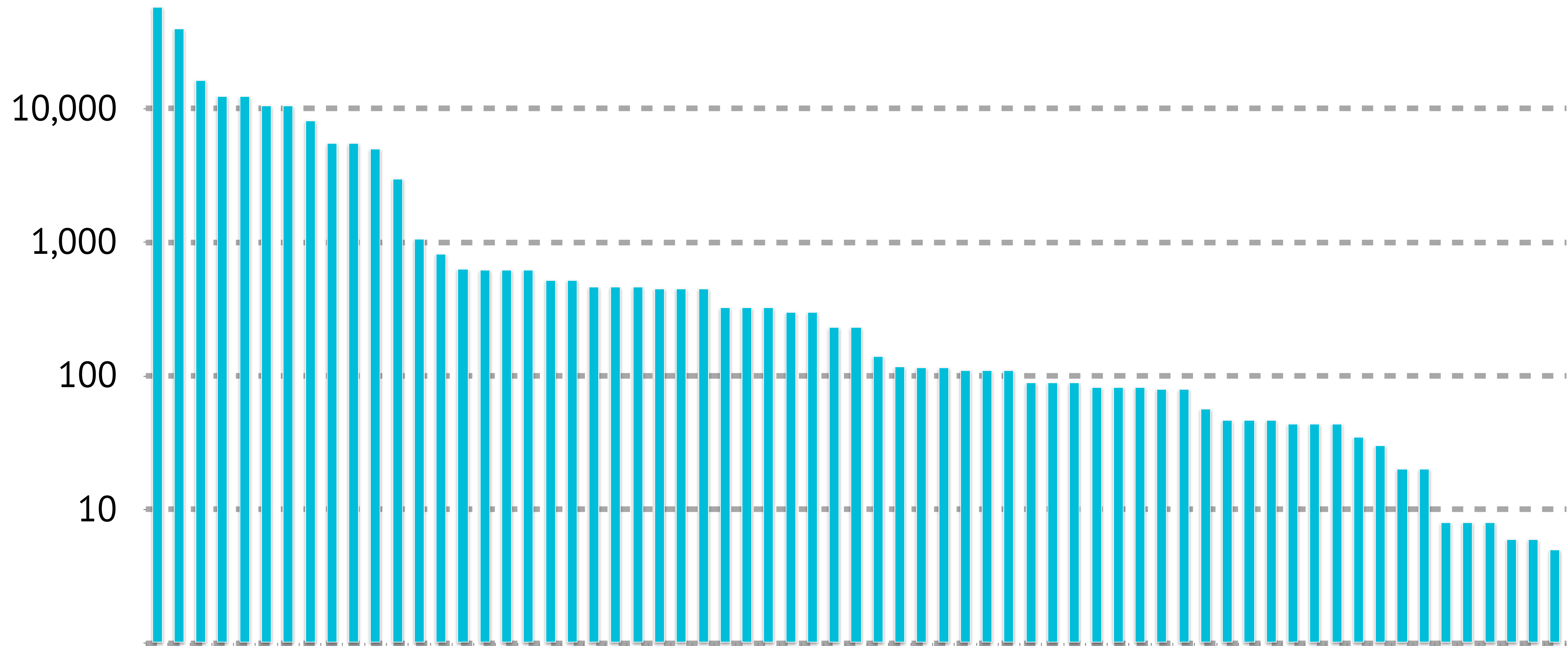
Relative Likelihood



Relative Severity



Joint Ranking



 @moyerma
 @evan2645

Findings

 @moyerma
 @evan2645

 SPIRE

Finding: Add Rate Limiting in spire-server

- SPIRE server should have some rate limiting controls:

The screenshot shows a GitHub pull request page for the repository 'spiffe / spire'. The pull request title is 'Add ratelimiter for node api #577'. It is marked as 'Merged' and was merged by 'evan2645' on Sep 11. The pull request description states: 'This commit adds ratelimiting logic for calls to the node api. It uses client IP as the ratelimiting key, and will drop log lines if/when a client is affected by the limiting. Signed-off-by: Evan Gilman evan@scytale.io'. The pull request was requested for review from several users: 'amartinezfayo', 'azdagron', 'MarcosDY', 'martincapello', and 'walmav'. The commit hash is 'e8e4c18'.

Finding: Eliminate CSR Parsing

- SPIFFE could drop CSR parsing altogether.
 - CSR (PKCS #10) format is complex.
 - CSR parsers are in general are likely less exercised than certificate parsers.
 - Security of SPIFFE doesn't rely on its signing guarantees.

Finding: No Big Surprises

- Overall, the design of the system enables the desired security properties.
- Doesn't mean there aren't issues we overlooked, but this is a good sign.

Scrutinizing SPIRE to Sensibly Strengthen SPIFFE Security

Matt Moyer, Evan Gilman



 @moyerma

 @evan2645