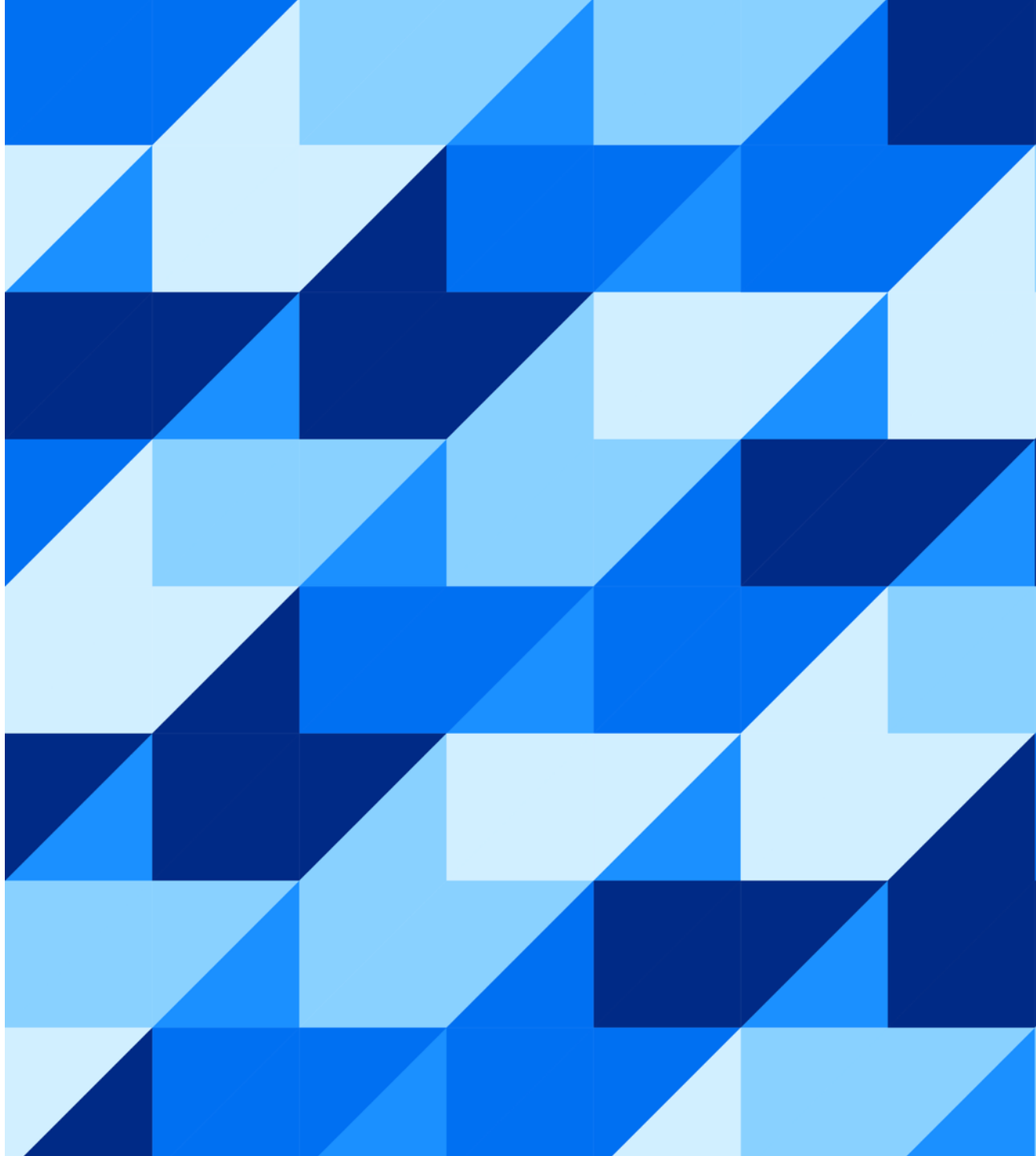
SAP BTP

Look Ma, No Secrets!

Georgi Lozev, Radoslav Tomov
SAP Labs Bulgaria



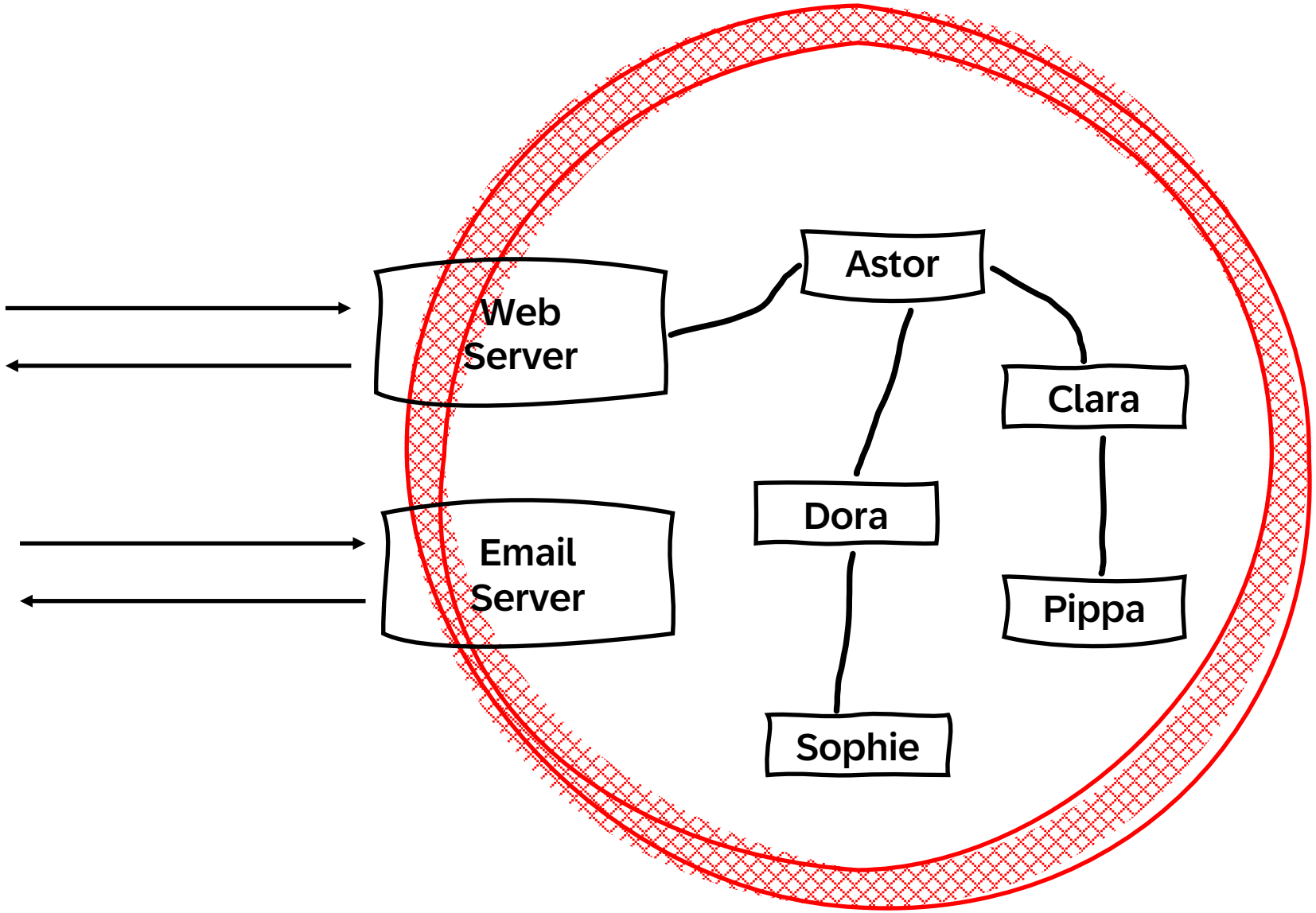
Agenda

- **The secret zero Problem**
- **SPIFFE & SPIRE as a solution**
- **Use case**
- **Q&A**

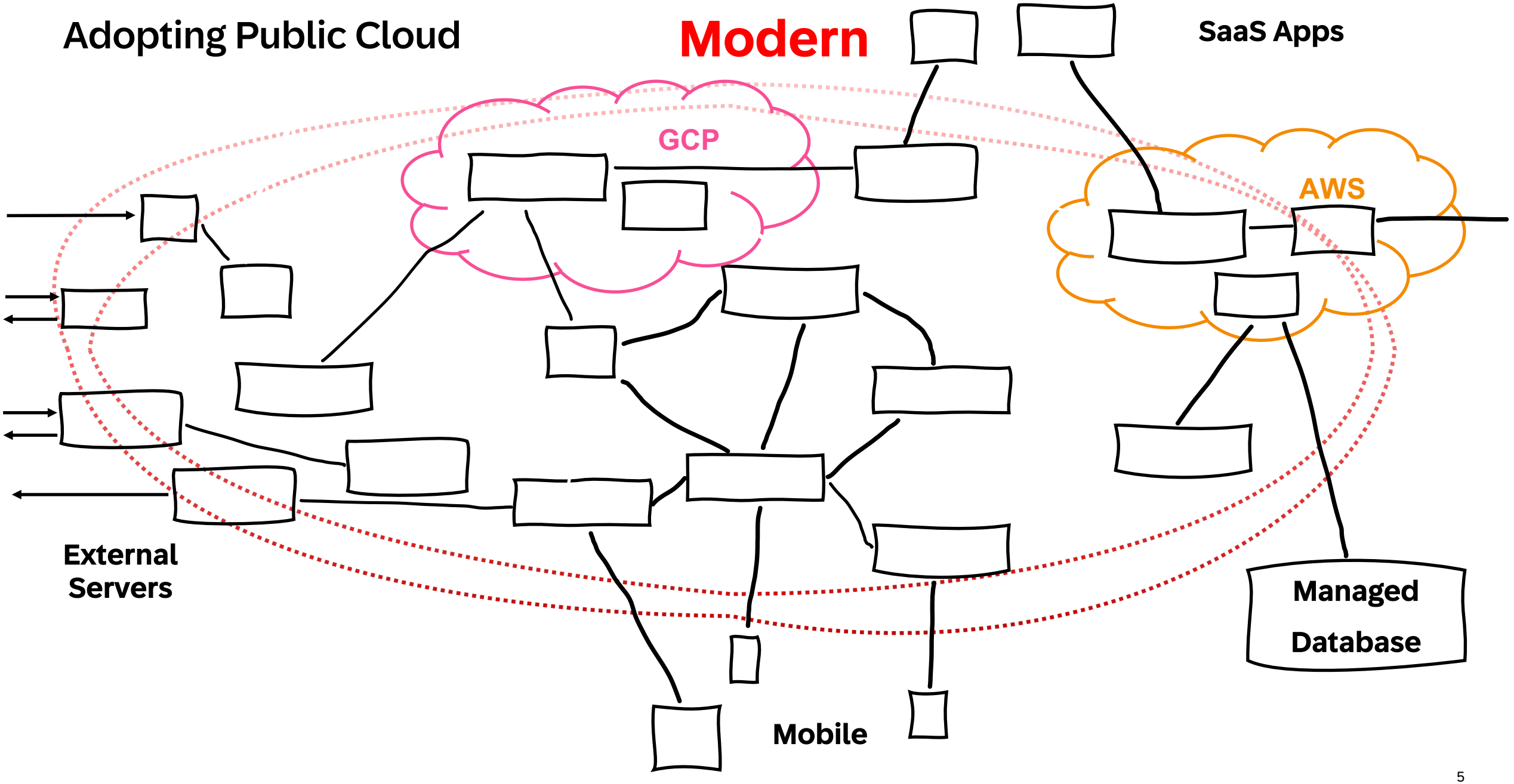
The secret zero Problem

Network Used to Be Friendly

Traditional

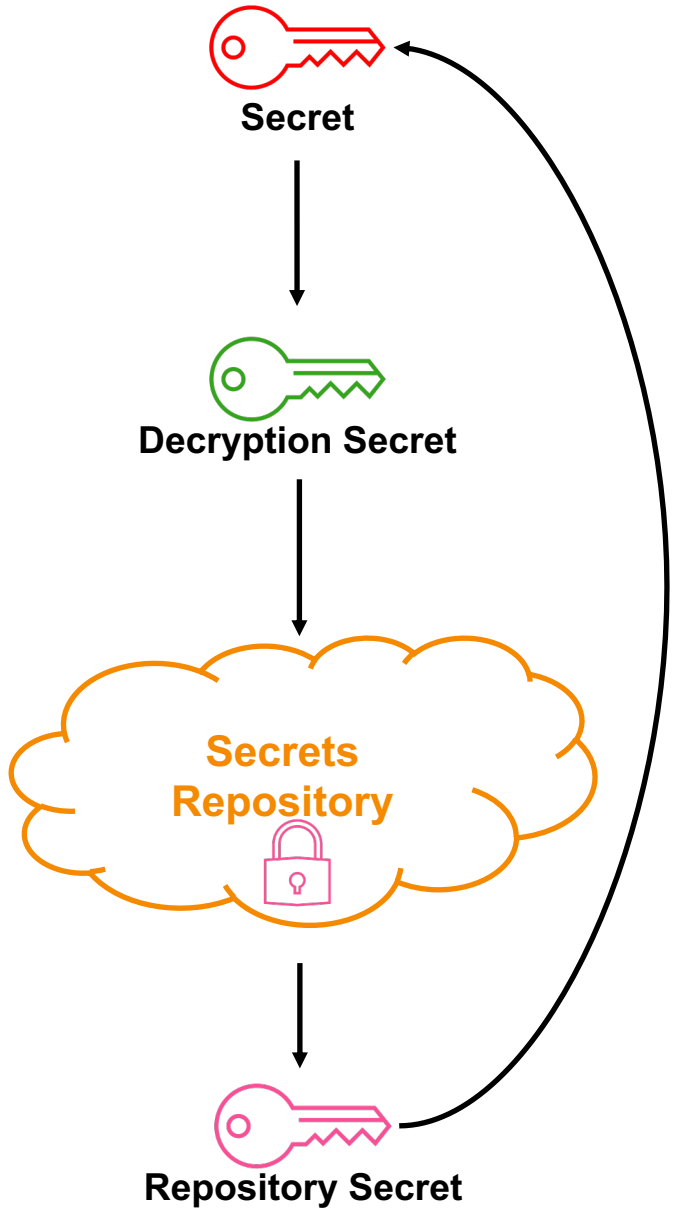


Adopting Public Cloud



Secrets as a Solution

- **Enabling access control to a database or service requires a secret**
- **That secret can be protected be with encryption, but then you still need to worry about a secret decryption key**
- **The decryption key could be put into a secrets repository, but then you still need a secret to access the secrets repository**
- **Ultimately, protecting access to one secret results in a new secret you need to protect**



Secrets as a Solution

Come with Challenges

- **Who generates the certificates and passwords, and how?**
- **How are they securely distributed to the applications that need them?**
- **How is access to private keys and passwords restricted?**
- **How are these secrets stored such that they don't leak into logs or backups?**
- **What happens when a certificate expires, or a password must be changed? Is the process disruptive?**
- **How many of these tasks necessarily involve a human operator?**
- **…**

SPIFFE & SPIRE as a Solution

SPIFFE and SPIRE

***S**ecure **P**roduction **I**dentity **F**ramework **F**or **E**veryone* defines a set of interfaces (APIs and docs) for proving, validating, and obtaining workload identity

***S**ecure **P**roduction **I**dentity **R**untime **E**nvironment* implements the SPIFFE interfaces and creates a toolchain for establishing trust between software systems

https://spiffe.io/
https://www.cncf.io/projects/spiffe/

SPIFFE

Overview

SPIFFE in One Sentence

Spec defining short lived cryptographic identity documents, called SVIDs, via simple API

Digital Passport

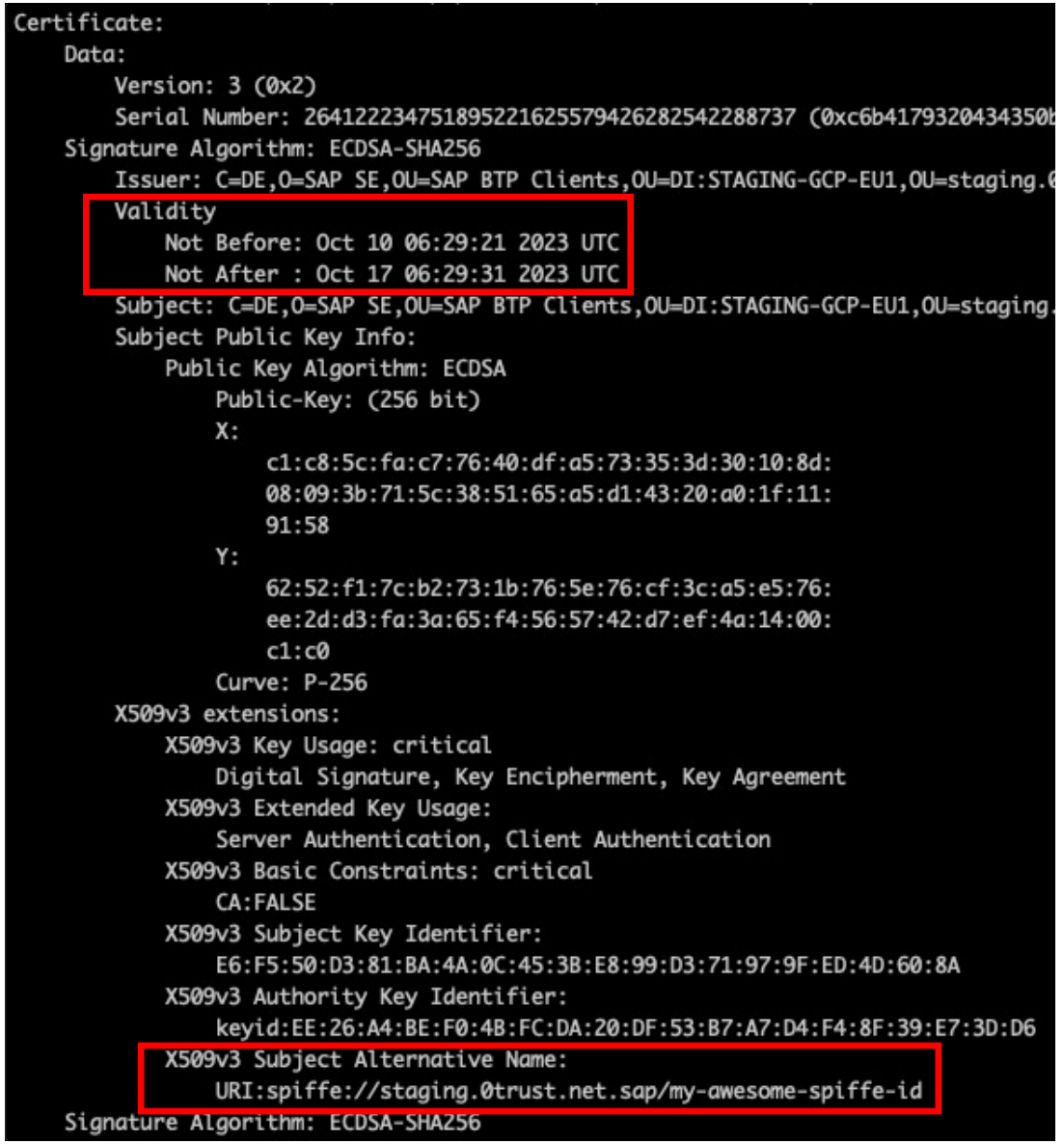
- Spec defining short lived cryptographic **identity documents**, called SVIDs, via simple API

Real World Identity Document

The SVID

- Spec defining **short lived** cryptographic identity documents, called **SVIDs**, via simple API

  – SPIFFE Verifiable Identity Document (SVID)
  – X.509 or JWT
  – includes The SPIFFE ID

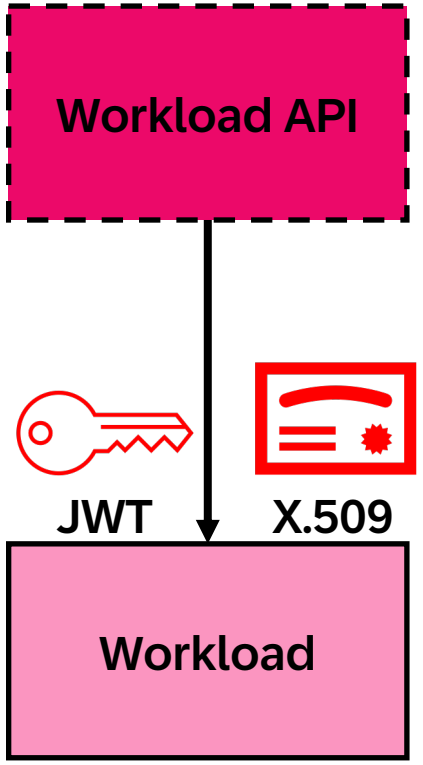


X.509 SVID

The Workload API

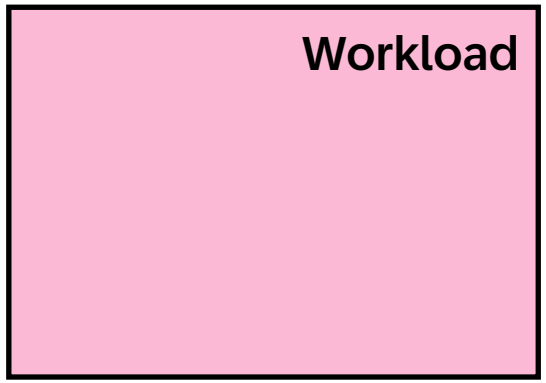
- Spec defining short lived cryptographic identity documents, called SVIDs, via simple API

  – serving (unauthenticated) workloads
  – streaming updates
  – solving the secret-zero problem

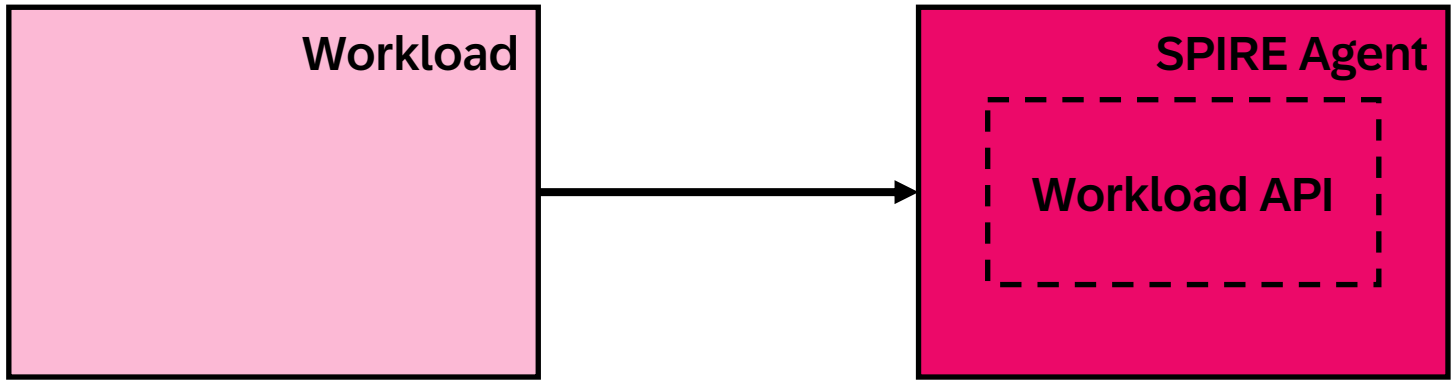


SPIRE
Basics

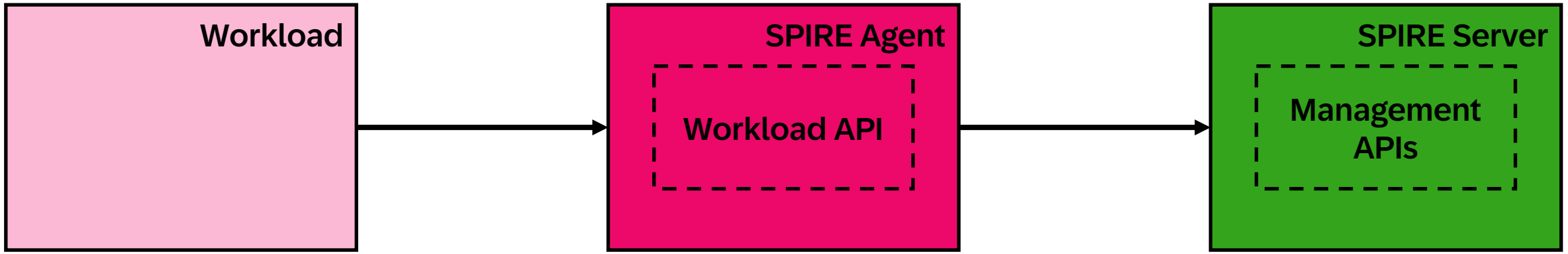
What is SPIRE ?



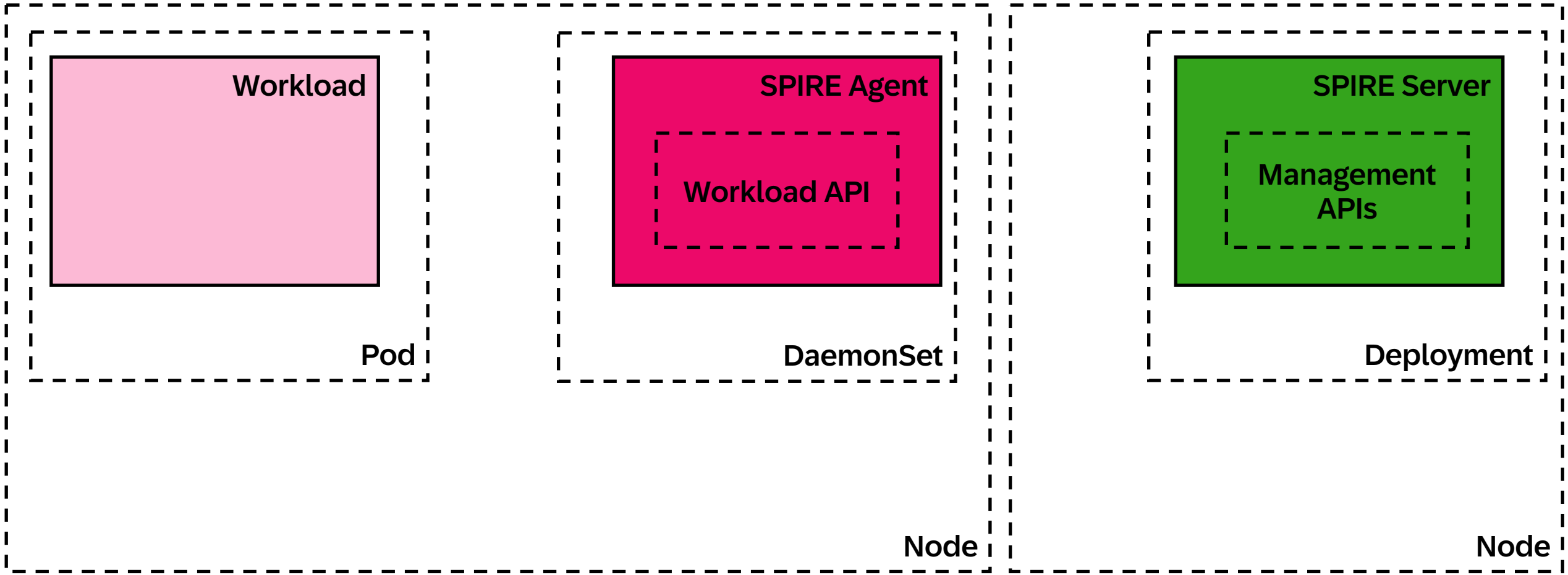
What is SPIRE ?



What is SPIRE ?

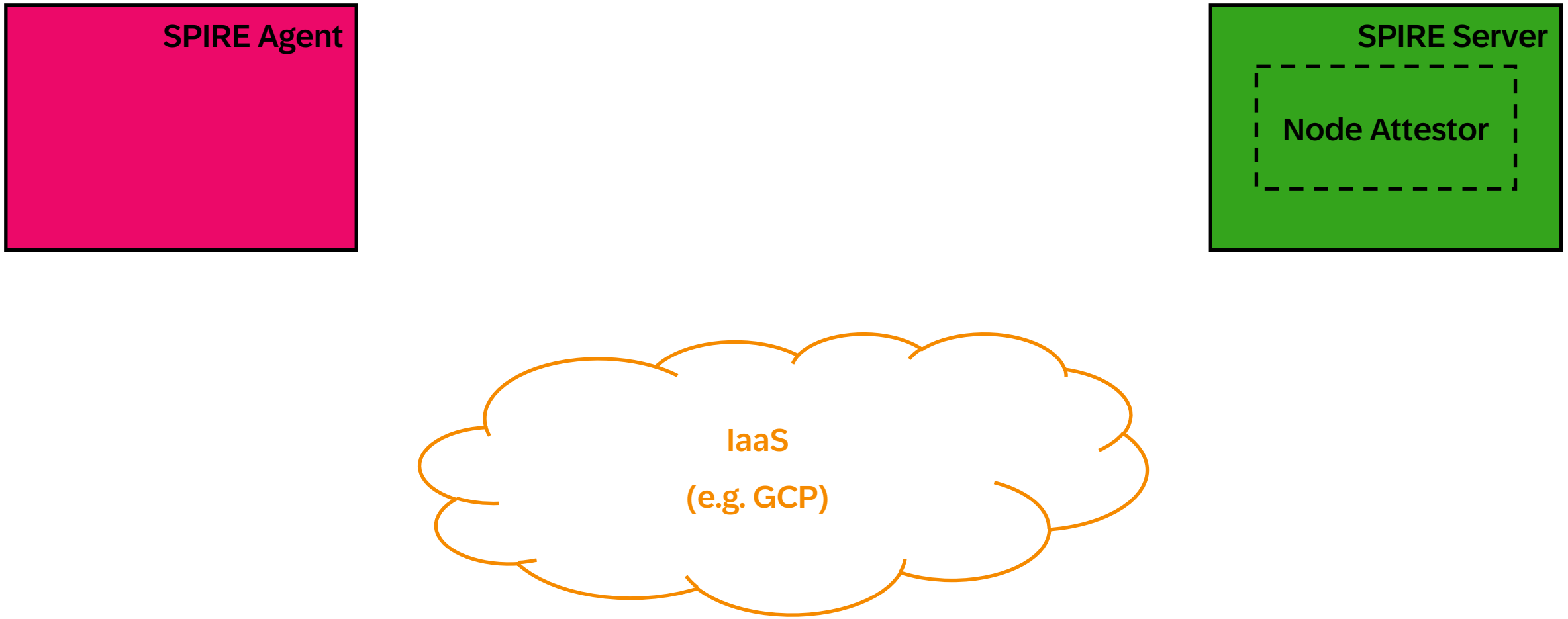


What is SPIRE ?
(in Kubernetes)

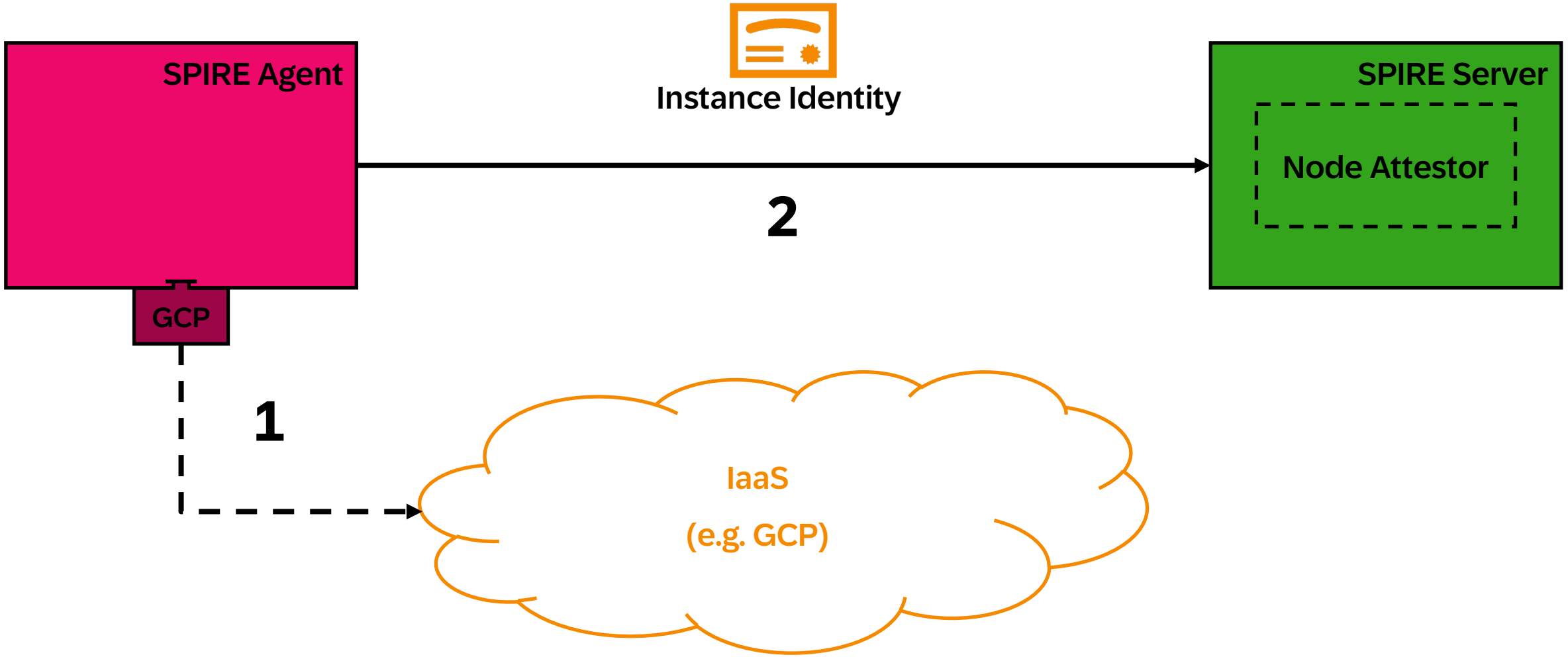


OK, BUT HOW ?

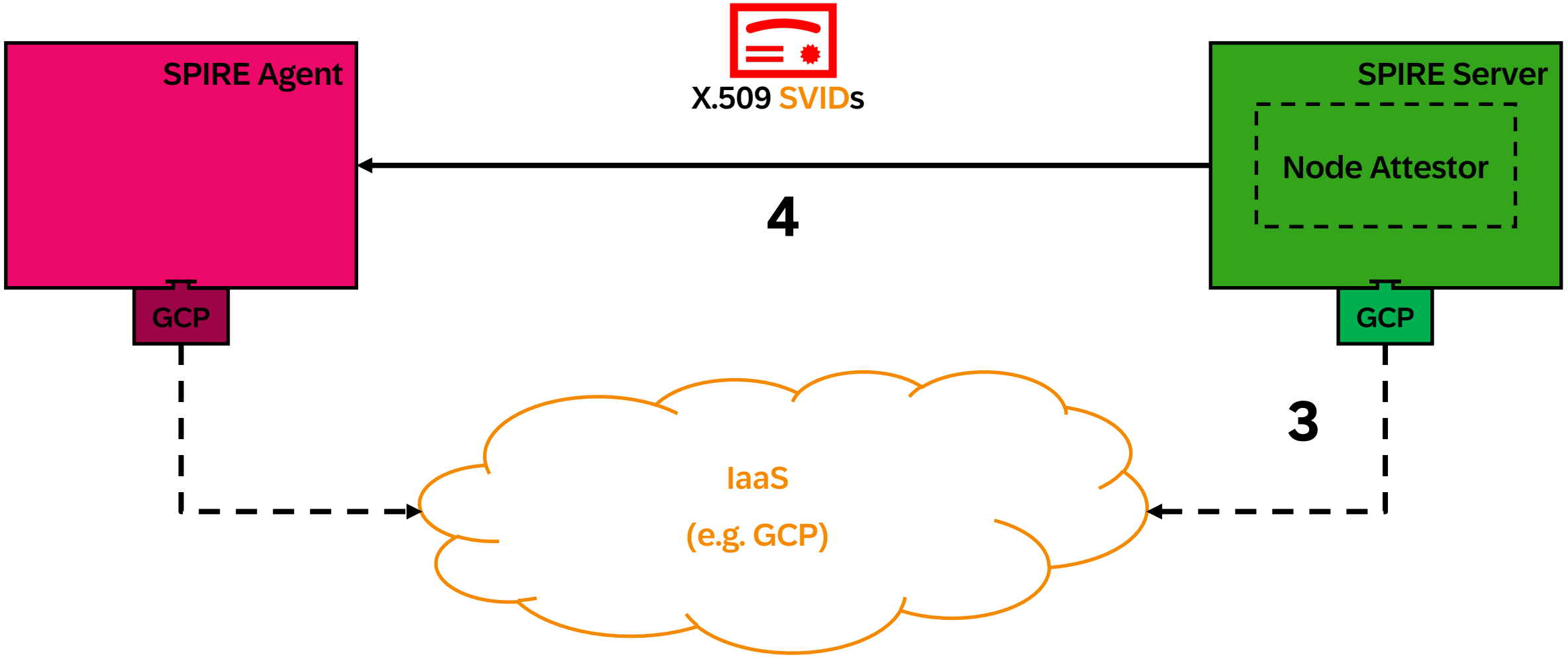
Node Attestation



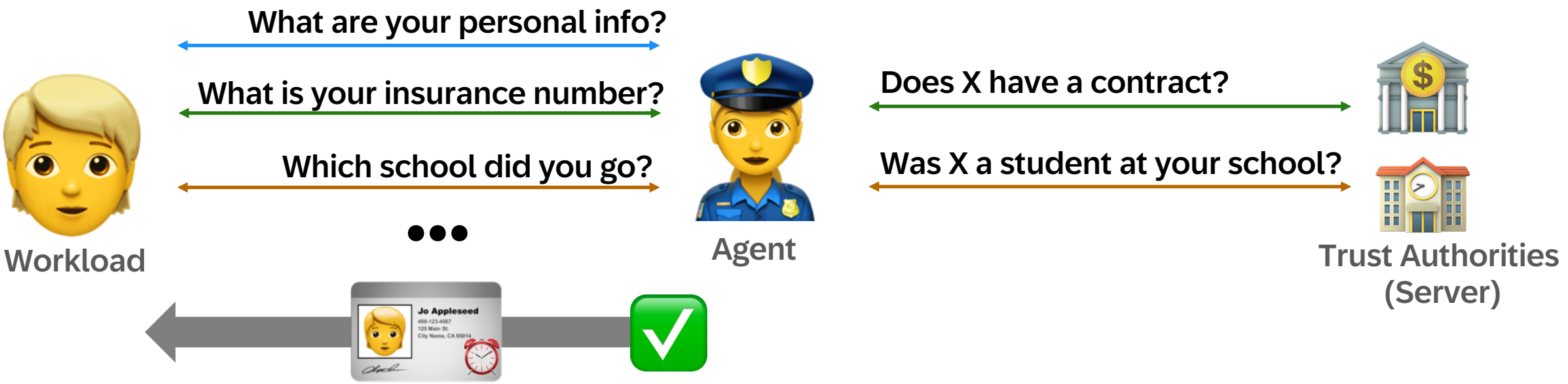
Node Attestation



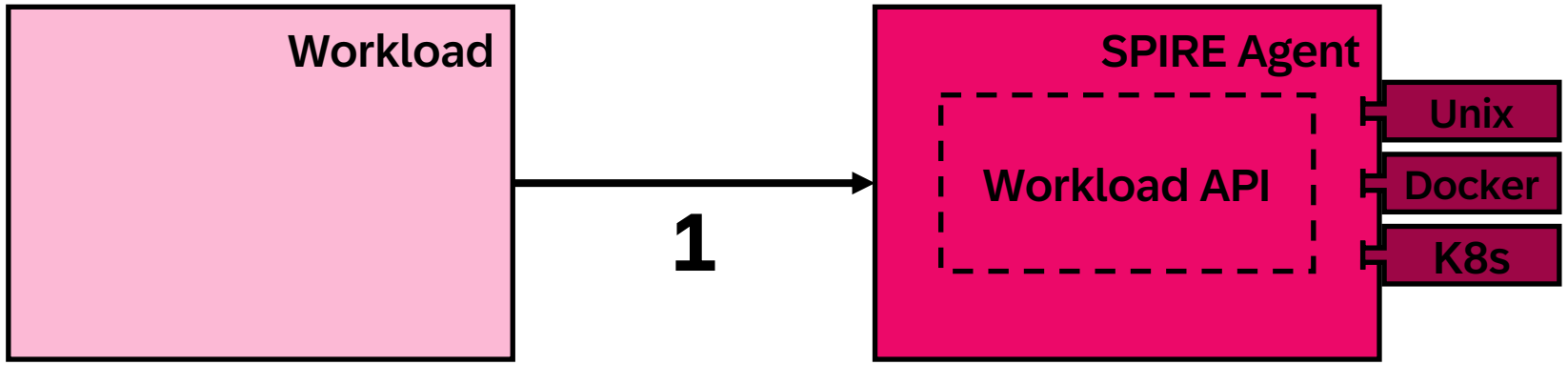
Node Attestation



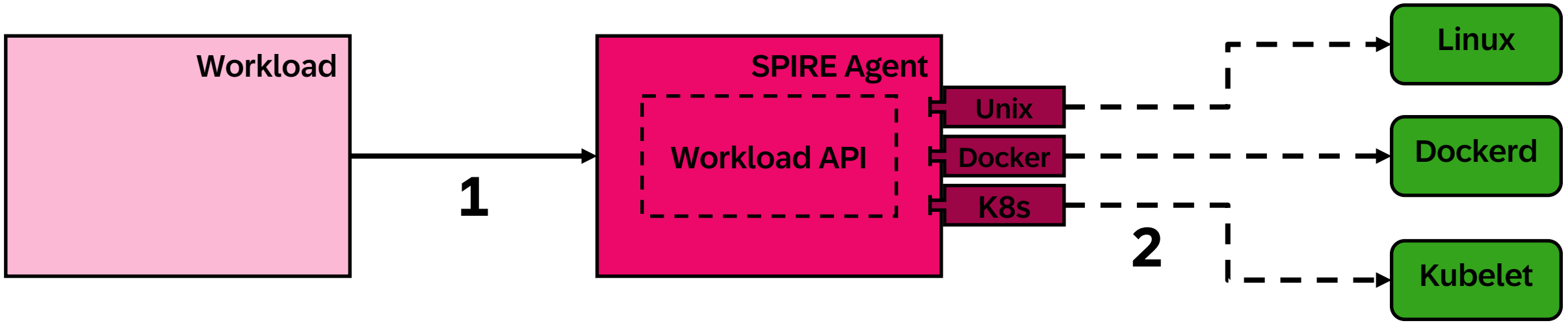
Establishes trust through attestation of multiple parameters



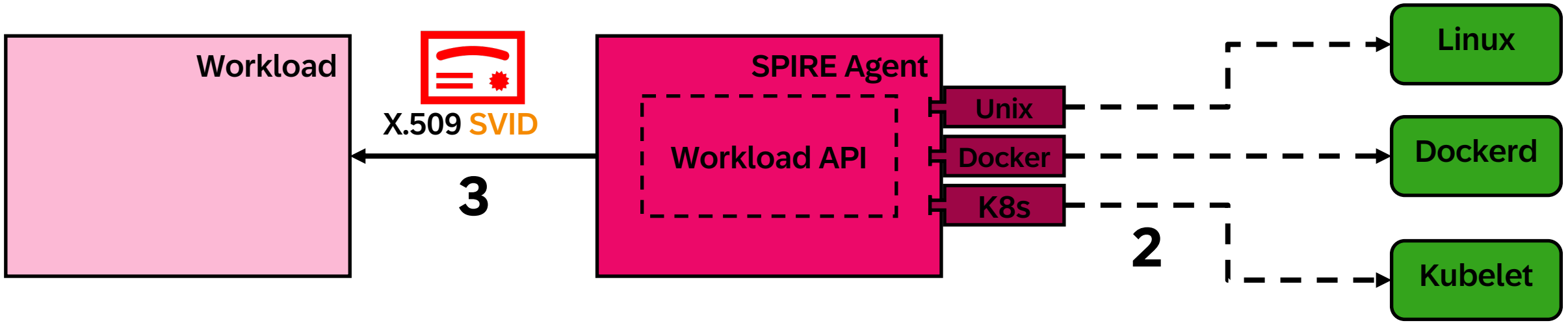
Workload Attestation



Workload Attestation



Workload Attestation



Use case

Setup Today

How to get a client certificate?

1. **Generate a Private Key:**

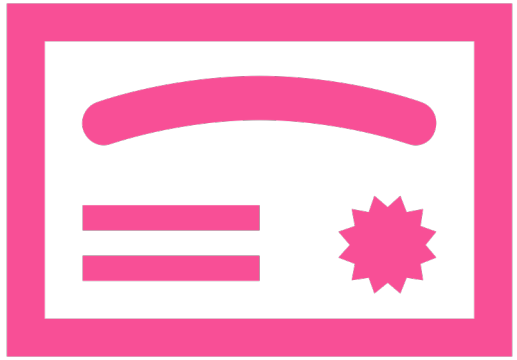
```
openssl genrsa -out client.key 2048
```

2. **Generate a Certificate Signing Request (CSR)**

```
openssl req -new -key client.key -out client.csr \
    -subj 'SOME SUBJECT THAT FITS THE POLICY'
```

3. **Submit the CSR to the Certificate Authority (CA)**

```
curl -X POST 'api.ca-authority/v1/certificate' \
  -H 'Authorization: Bearer <SECRET-1> ' \
  -d '{
        "csr": {
            "value": "..."
        }
      }'
```



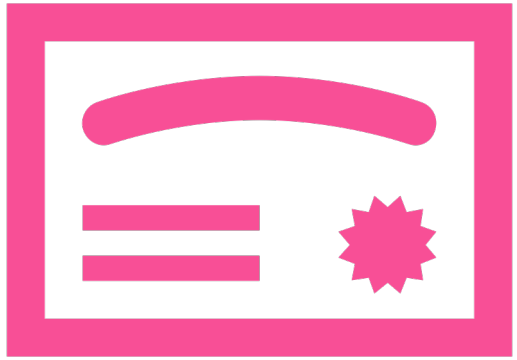
How to get a client certificate?

4. **CA signs the CSR and returns the chain**

```
extract_client_certificate_from_chain(cert_chain)
```

5. **Read the signed certificate**

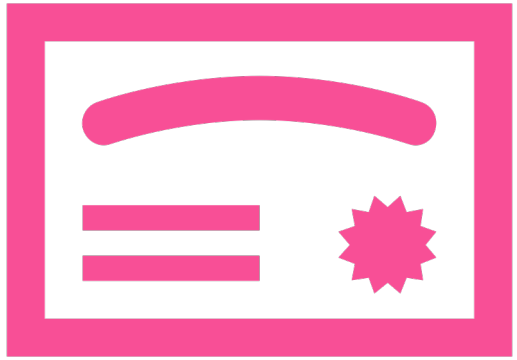
```
openssl x509 -text -noout -in client.pem
```



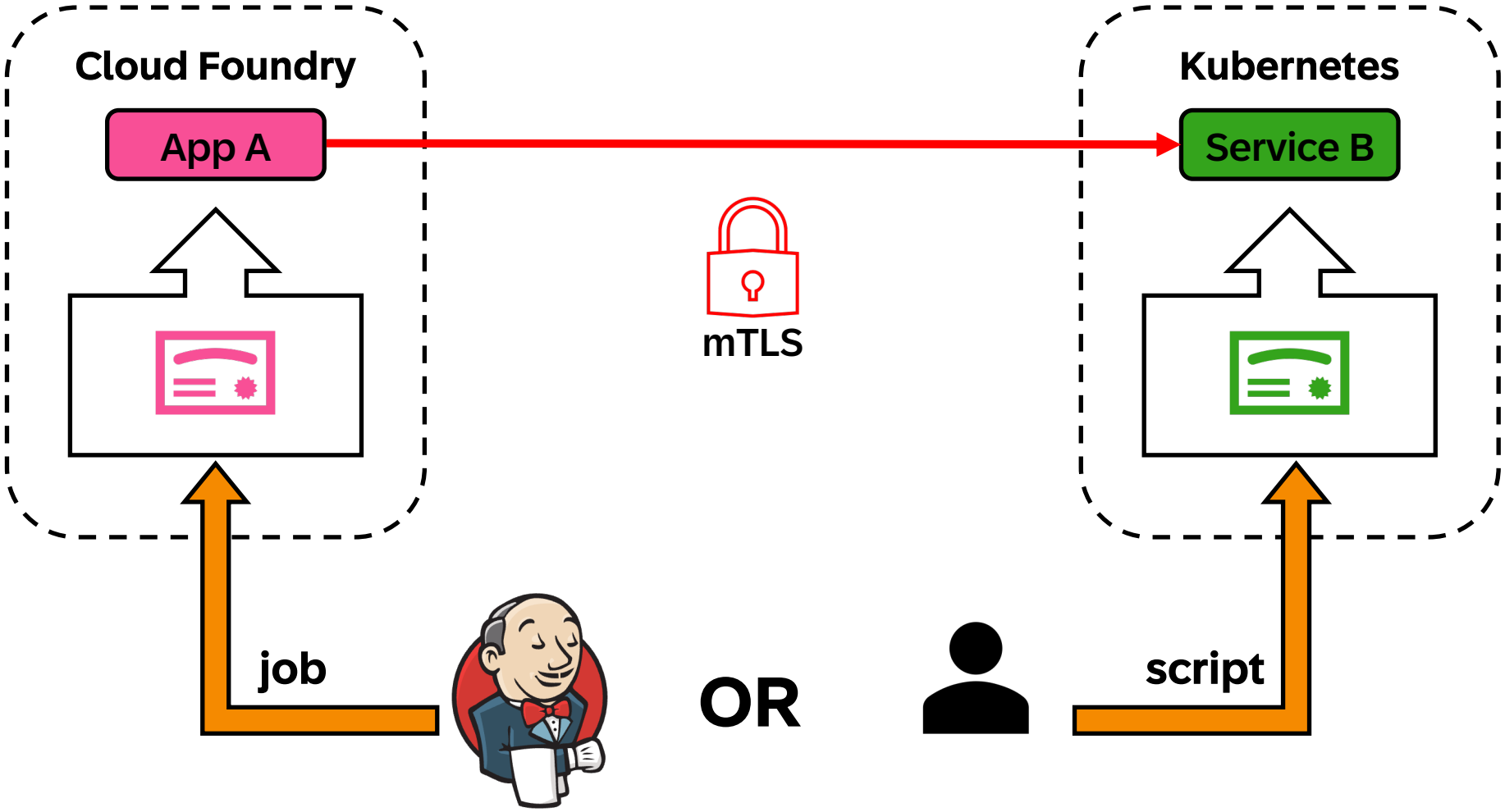
How to get a client certificate?

In a Nutshell

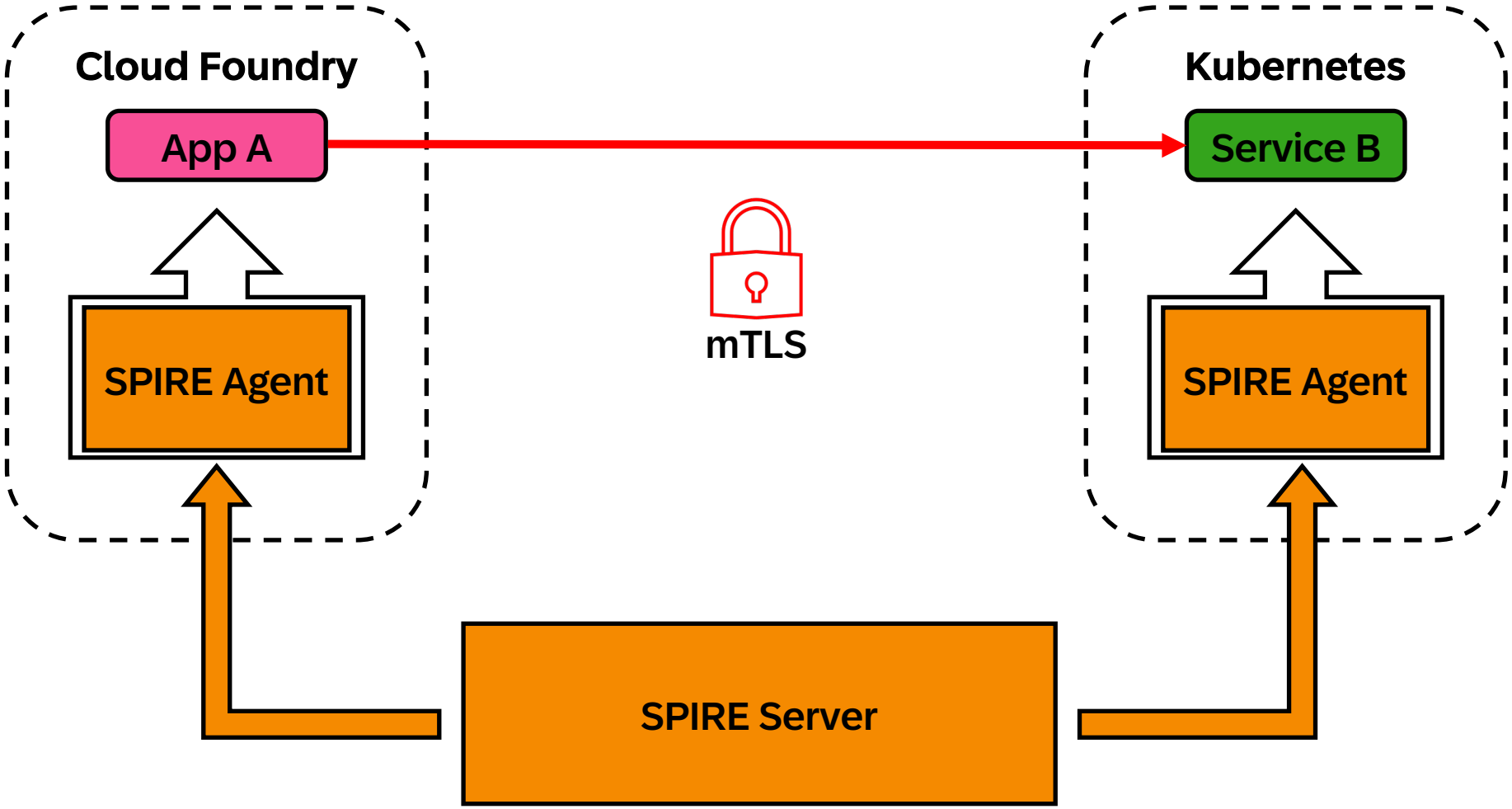
- **Mutli-Step Process**
- **Secrets are Involved**
- **Many Decisions with Impact on Security**



Setup Today



Adopting SPIRE



Key Takeaways

- **Security is a journey - go Zero Trust and Secure by Default**
- **Put your trust in Industry Standards, Tools and Processes**
- **You can get both - Security & Innovation Speed**

[1] Designing Data-Intensive Applications

[2] Ben Moseley and Peter Marks: "Out of the Tar Pit"

[3] Zero Trust Architecture

[4] The Bottom Turtle

Thank you!

Contact information:

- radoslav.tomov@sap.com
- georgi.lozev@sap.com