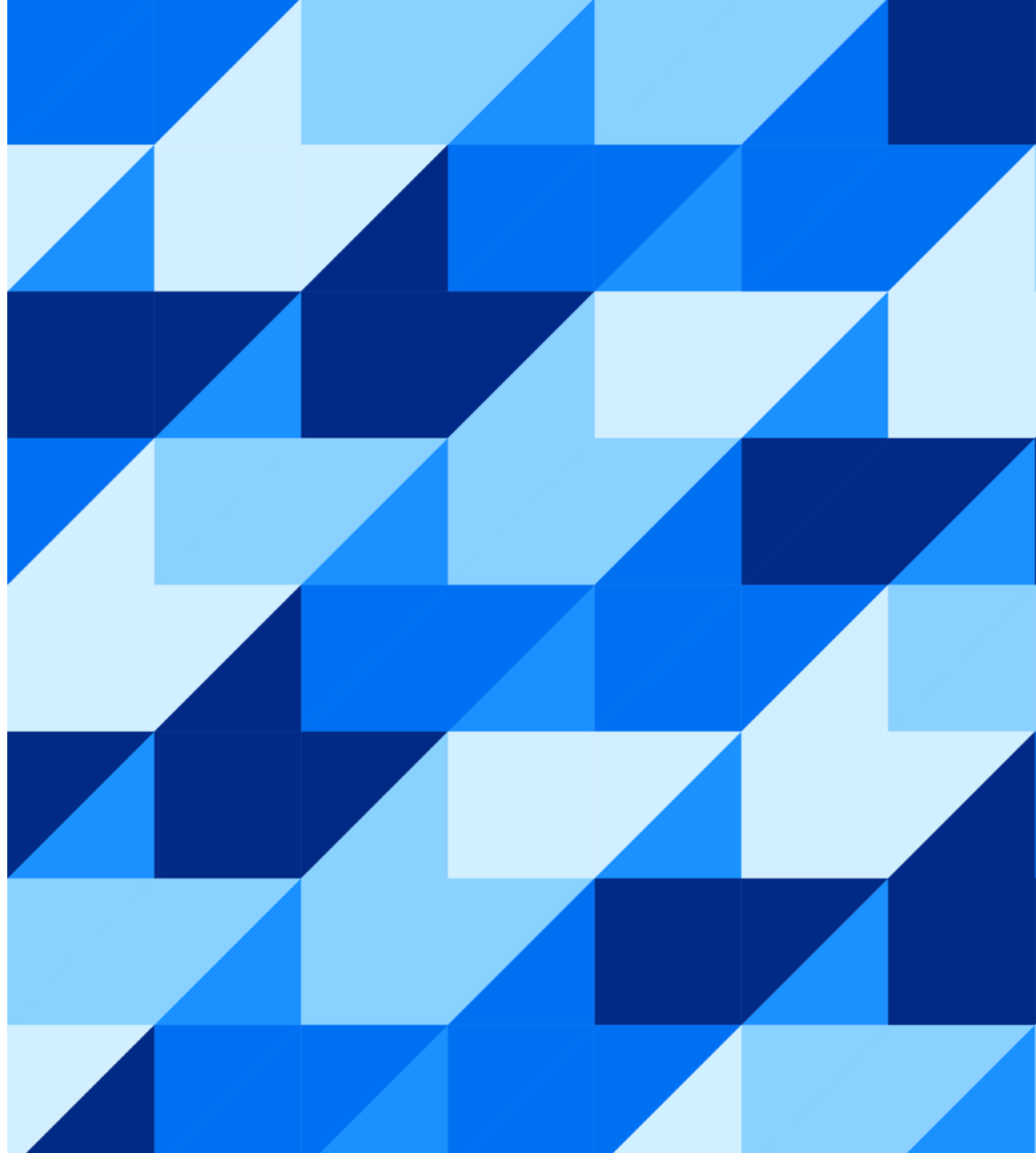


Look Ma, No Secrets!

Georgi Lozev, Radoslav Tomov
SAP Labs Bulgaria



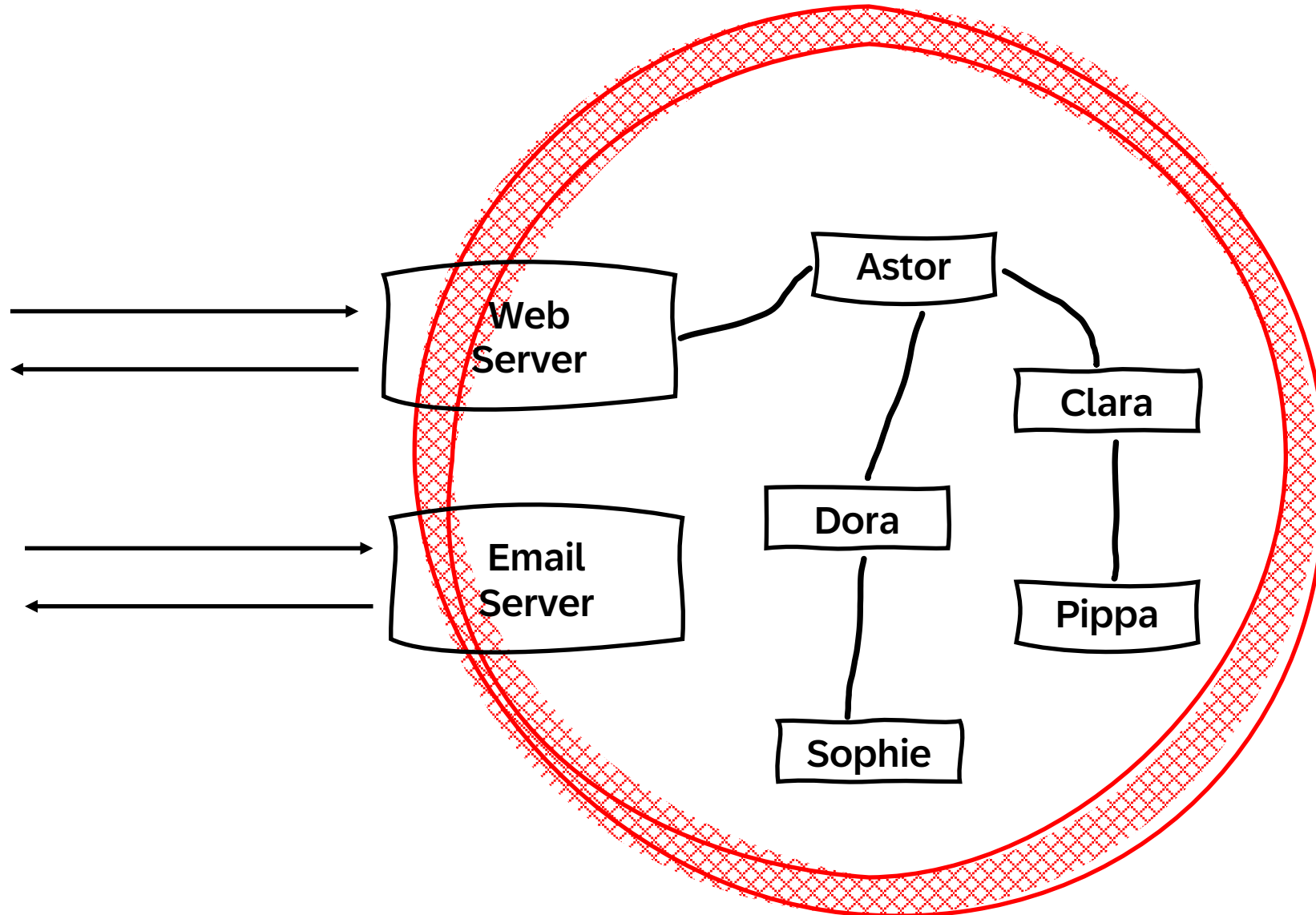
Agenda

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

The **secret** zero Problem

Network Used to Be Friendly

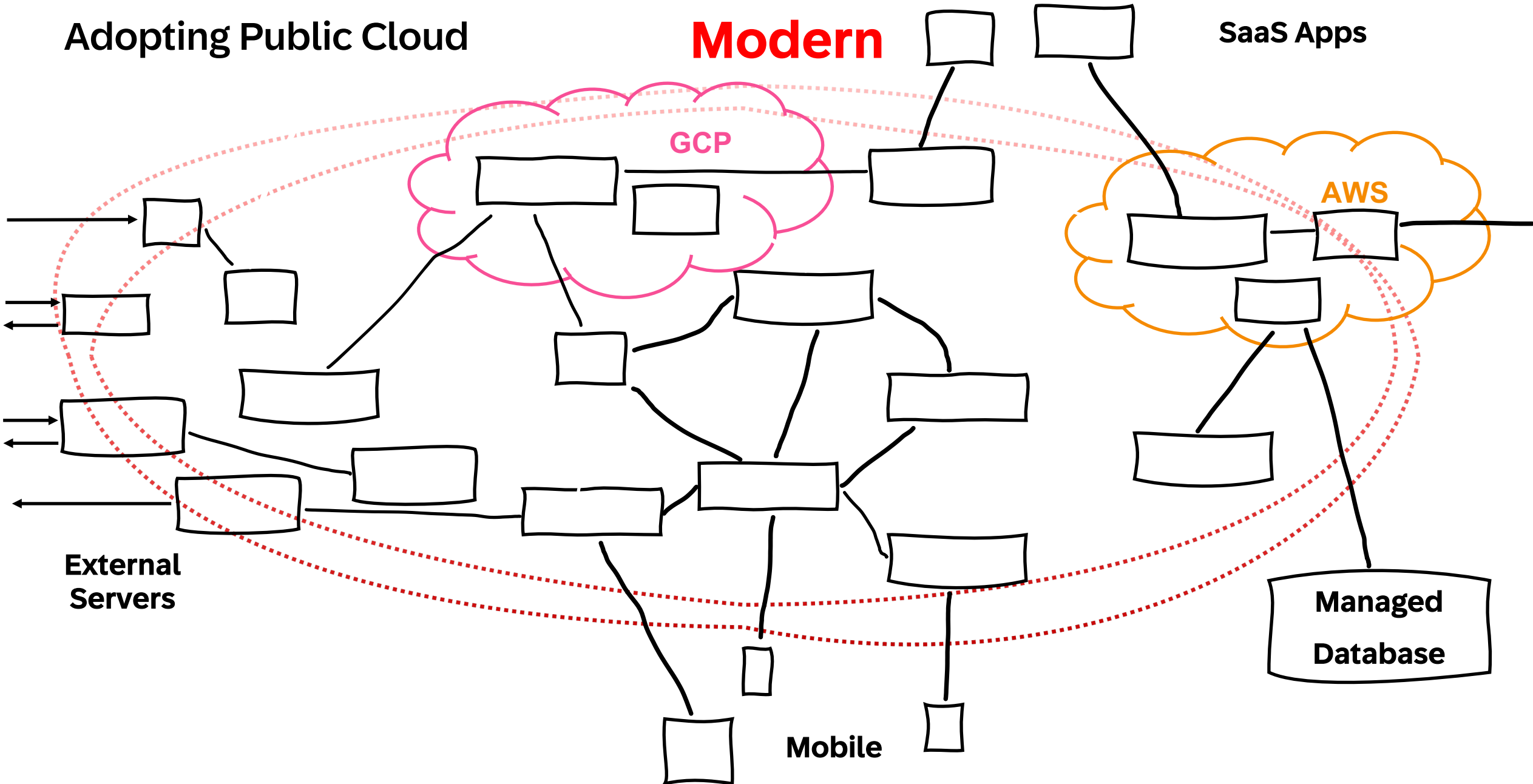
Traditional



Adopting Public Cloud

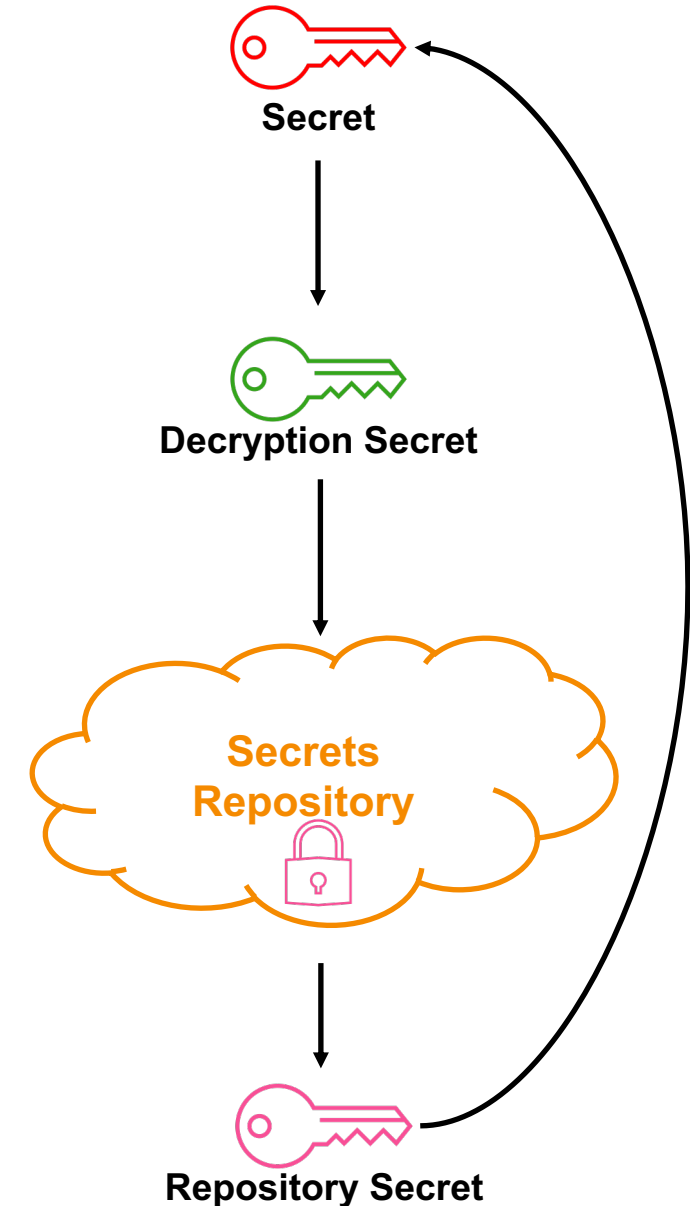
Modern

SaaS Apps



Secrets as a Solution

- Enabling access control to a database or service requires a **secret**
- That **secret** can be protected 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?**
- ...



Kelsey Hightower ✓ @kelseyhightower · Nov 6

I'm actually a fan of Kubernetes secrets: named byte arrays that can be encrypted at rest, protected via RBAC, and distributed over TLS.

Unfortunately the name "secrets" causes confusion when people type cast secrets to mean credentials or any other form of sensitive data.

💬 18

↻ 34

❤ 334



SPIFFE & SPIRE as a Solution

SPIFFE and SPIRE



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



Secure *P*roduction *I*ntity *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

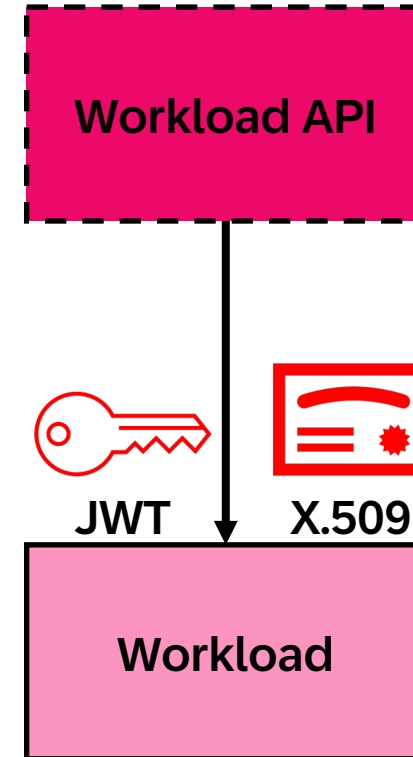
- Spec defining **short lived** cryptographic identity documents, called **SVIDs**, via simple API
 - SPIFFE Verifiable Identity Document (**SVID**)
 - X.509 or JWT
 - includes **The SPIFFE ID**

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 264122234751895221625579426282542288737 (0xc6b4179320434350b)
    Signature Algorithm: ECDSA-SHA256
    Issuer: C=DE,O=SAP SE,OU=SAP BTP Clients,OU=DI:STAGING-GCP-EU1,OU=staging.0
    Validity
      Not Before: Oct 10 06:29:21 2023 UTC
      Not After : Oct 17 06:29:31 2023 UTC
    Subject: C=DE,O=SAP SE,OU=SAP BTP Clients,OU=DI:STAGING-GCP-EU1,OU=staging.0
    Subject Public Key Info:
      Public Key Algorithm: ECDSA
      Public-Key: (256 bit)
      X:
        c1:c8:5c:fa:c7:76:40:df:a5:73:35:3d:30:10:8d:
        08:09:3b:71:5c:38:51:65:a5:d1:43:20:a0:1f:11:
        91:58
      Y:
        62:52:f1:7c:b2:73:1b:76:5e:76:cf:3c:a5:e5:76:
        ee:2d:d3:fa:3a:65:f4:56:57:42:d7:ef:4a:14:00:
        c1:c0
      Curve: P-256
    X509v3 extensions:
      X509v3 Key Usage: critical
        Digital Signature, Key Encipherment, Key Agreement
      X509v3 Extended Key Usage:
        Server Authentication, Client Authentication
      X509v3 Basic Constraints: critical
        CA:FALSE
      X509v3 Subject Key Identifier:
        E6:F5:50:D3:81:BA:4A:0C:45:3B:E8:99:D3:71:97:9F:ED:4D:60:8A
      X509v3 Authority Key Identifier:
        keyid:EE:26:A4:BE:F0:4B:FC:DA:20:DF:53:B7:A7:D4:F4:8F:39:E7:3D:D6
      X509v3 Subject Alternative Name:
        URI:spiffe://staging.0trust.net.sap/my-awesome-spiffe-id
    Signature Algorithm: ECDSA-SHA256
```

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

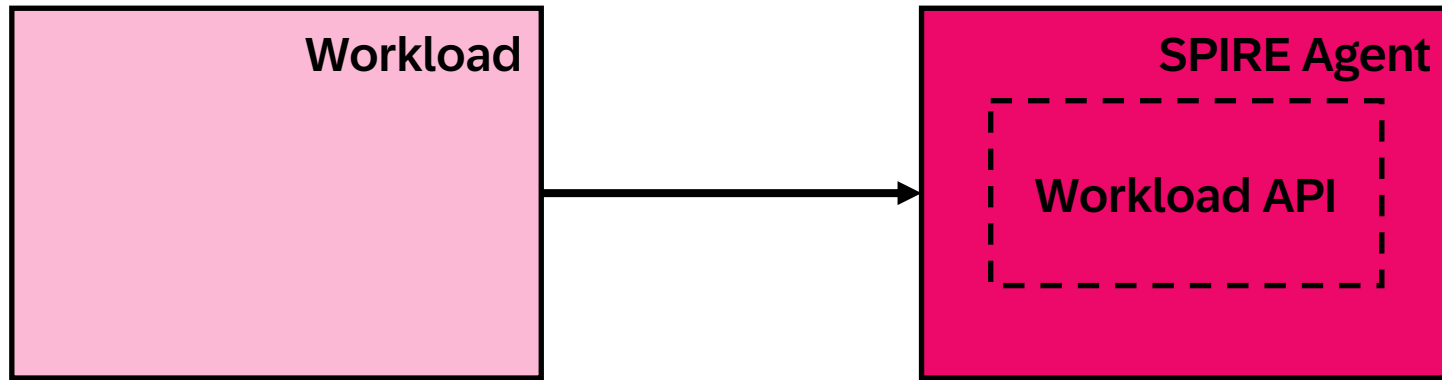


What is **SPIRE** ?

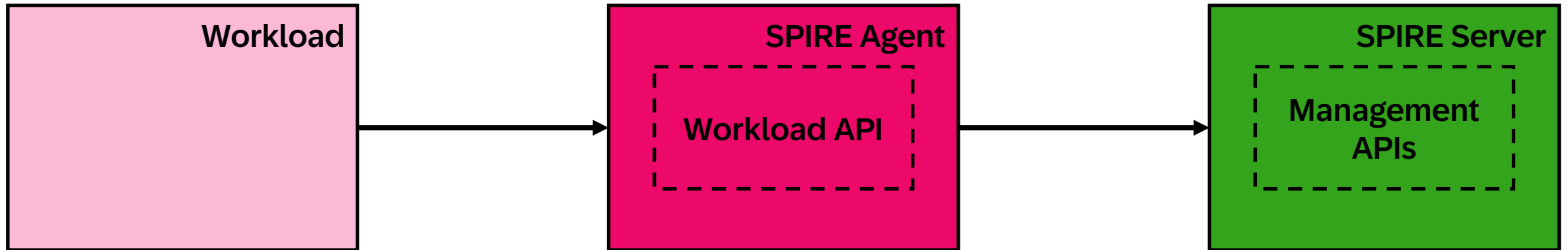


Workload

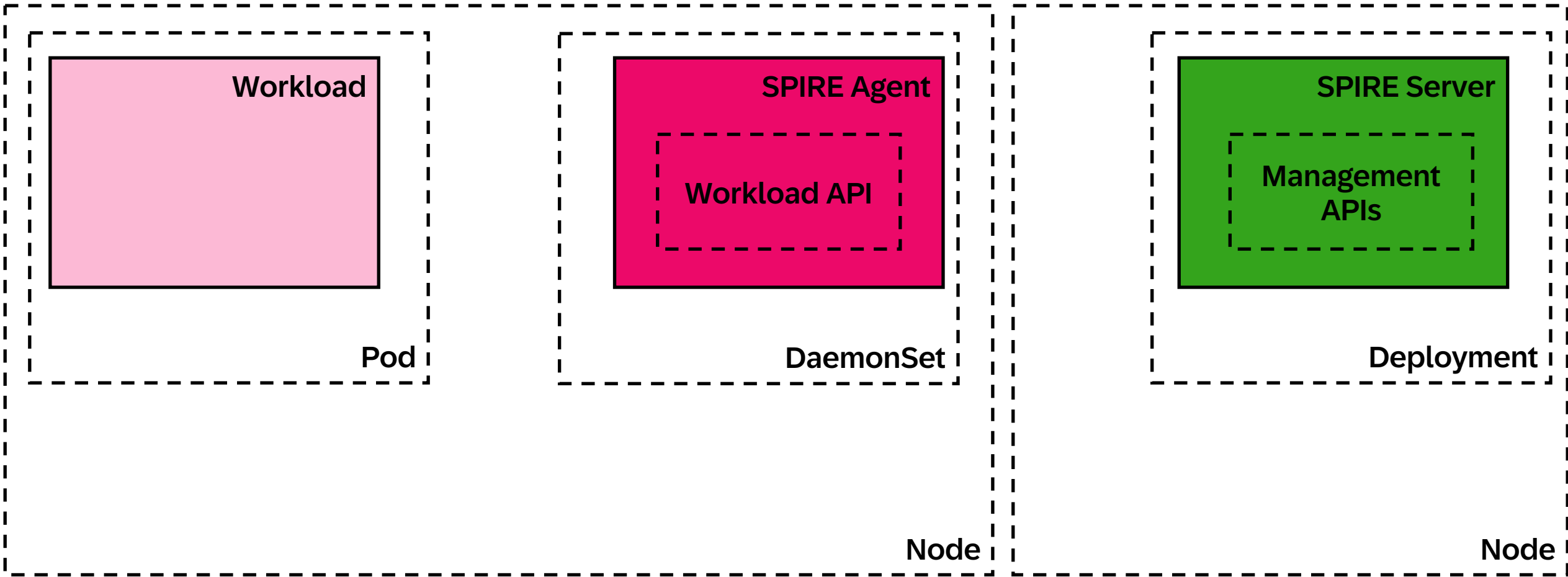
What is SPIRE ?



What is SPIRE ?



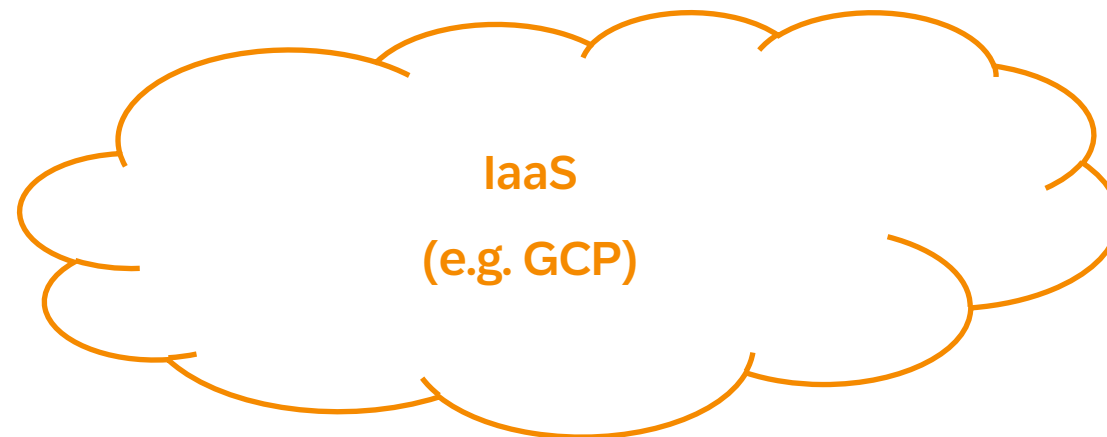
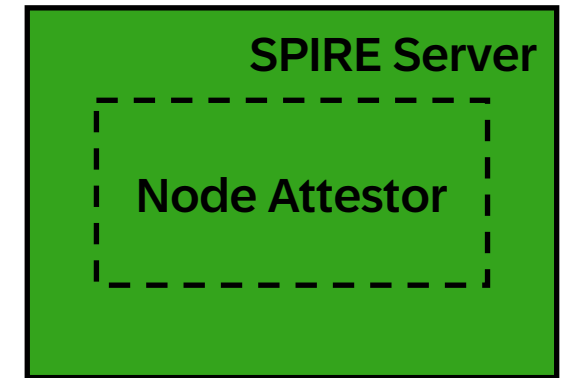
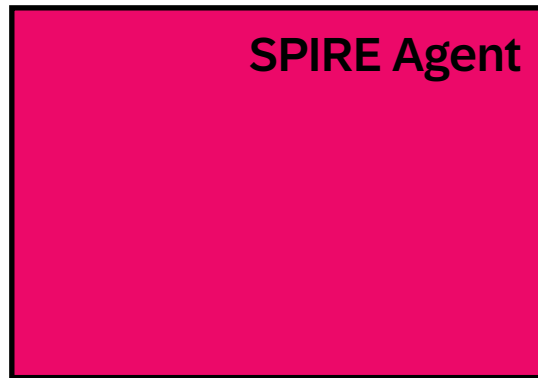
What is **SPIRE** ? (in Kubernetes)



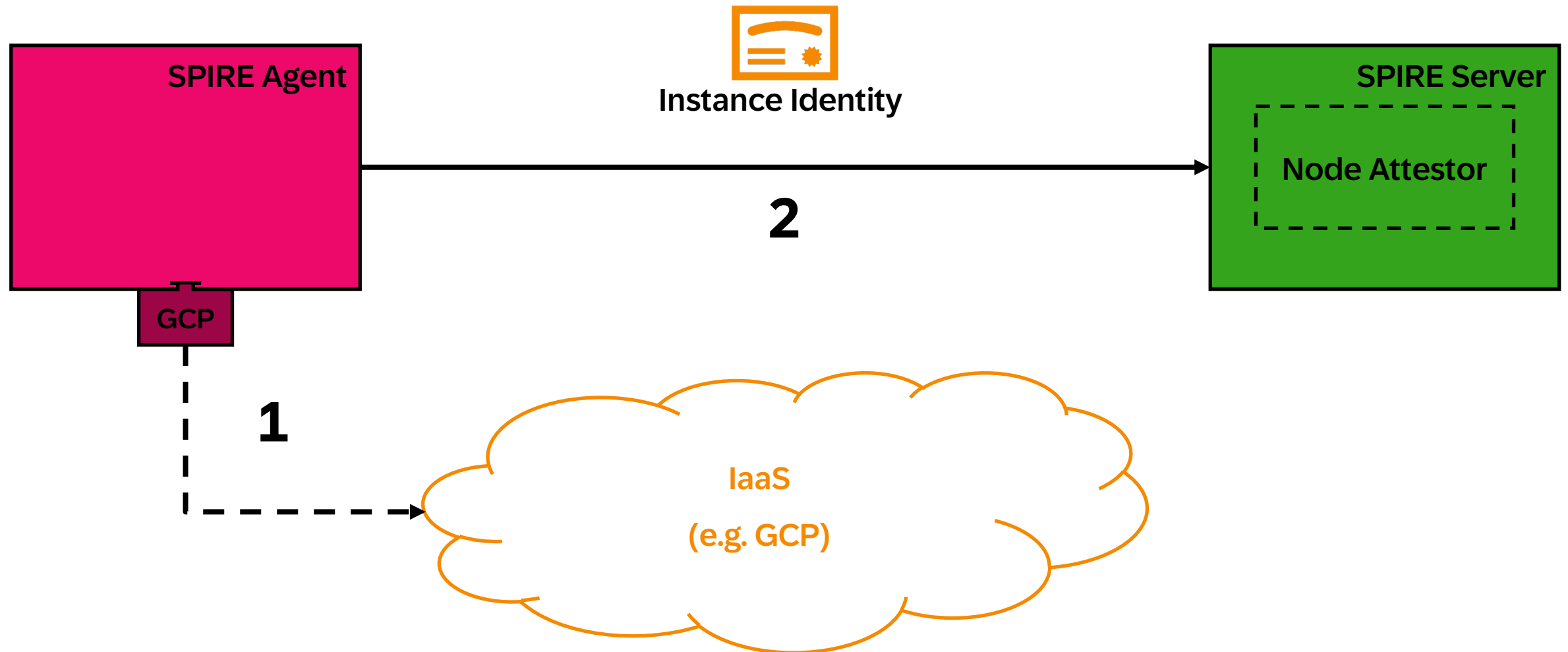
OK, BUT HOW ?



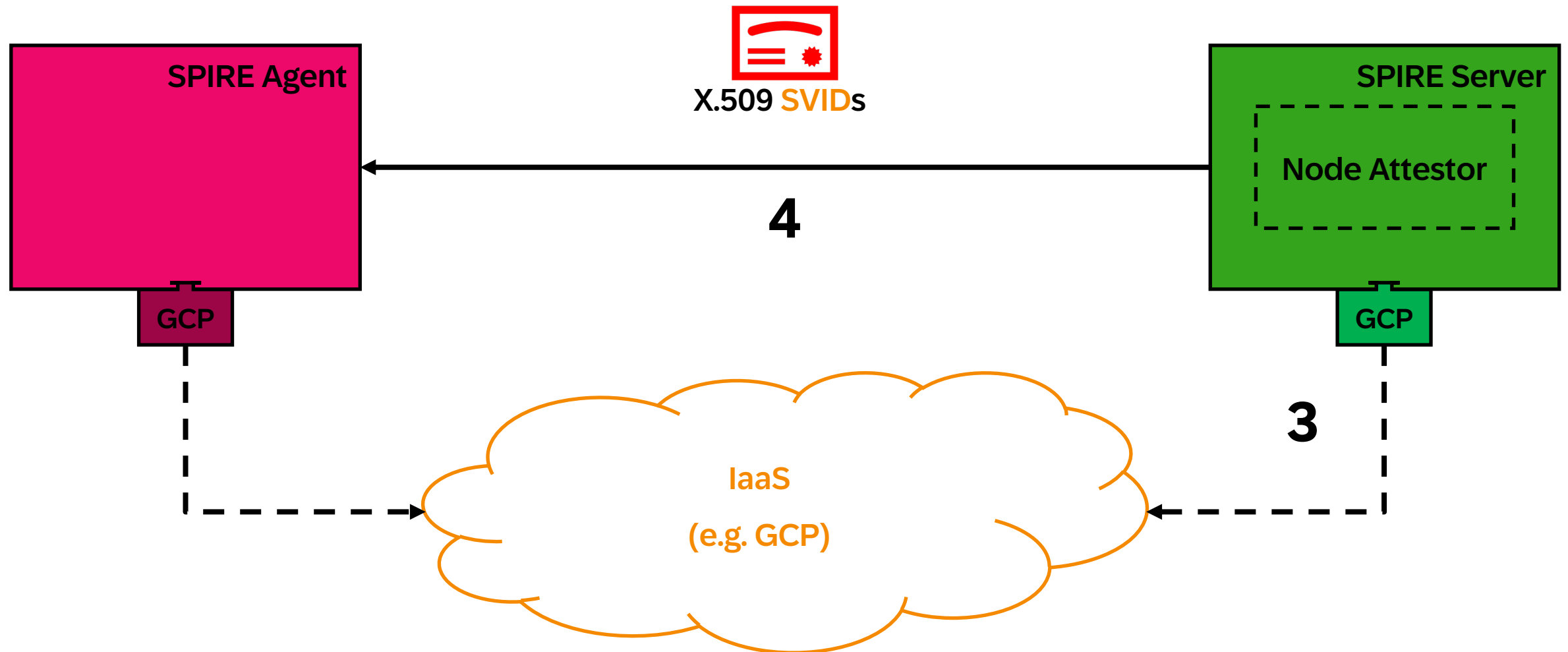
Node Attestation



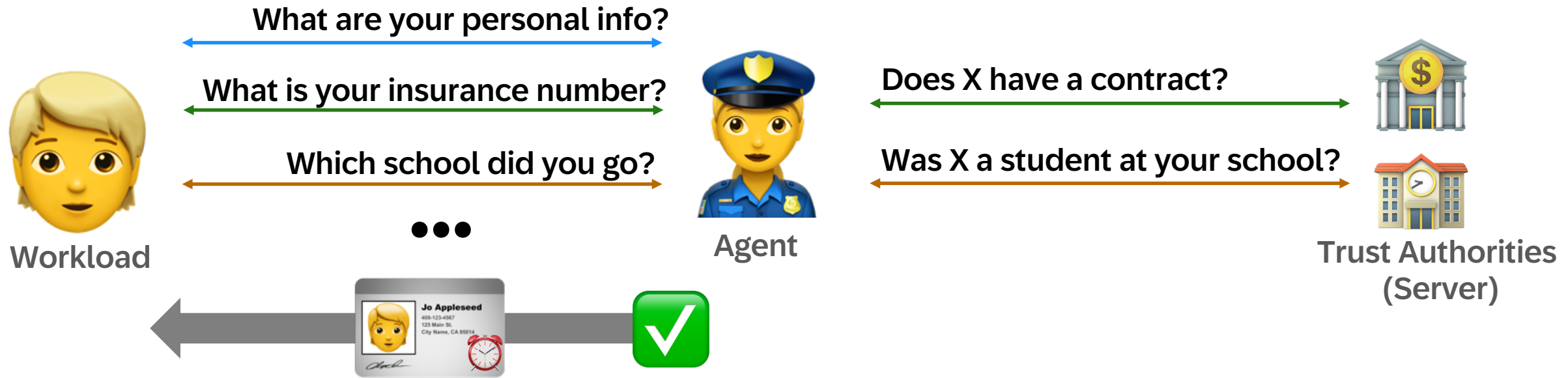
Node Attestation



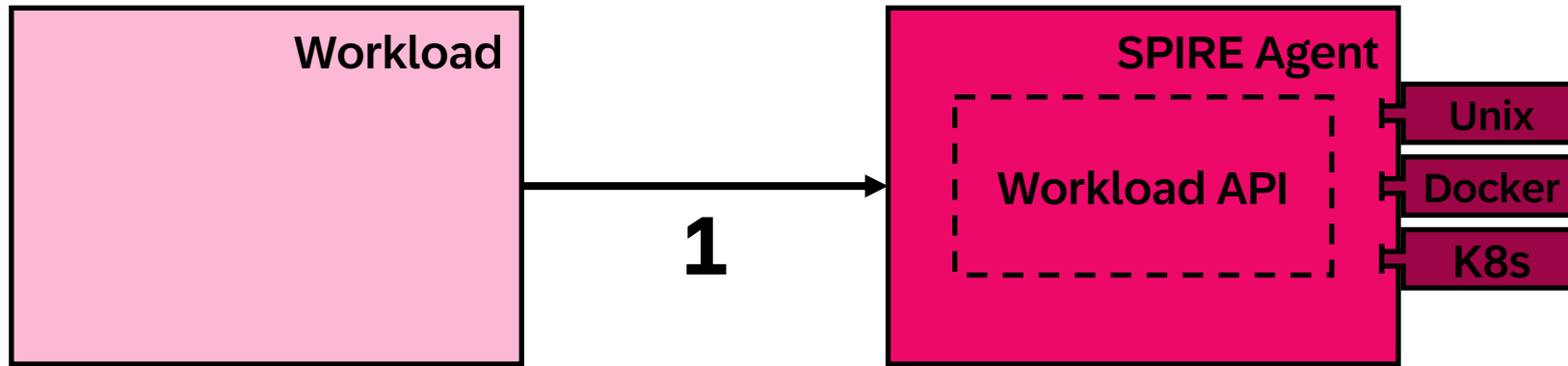
Node Attestation



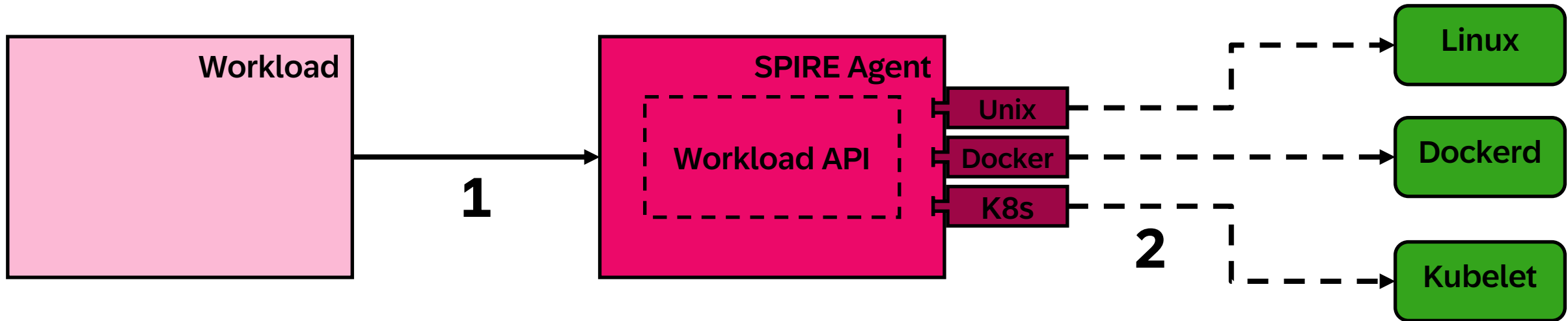
Establishes trust through attestation of multiple parameters



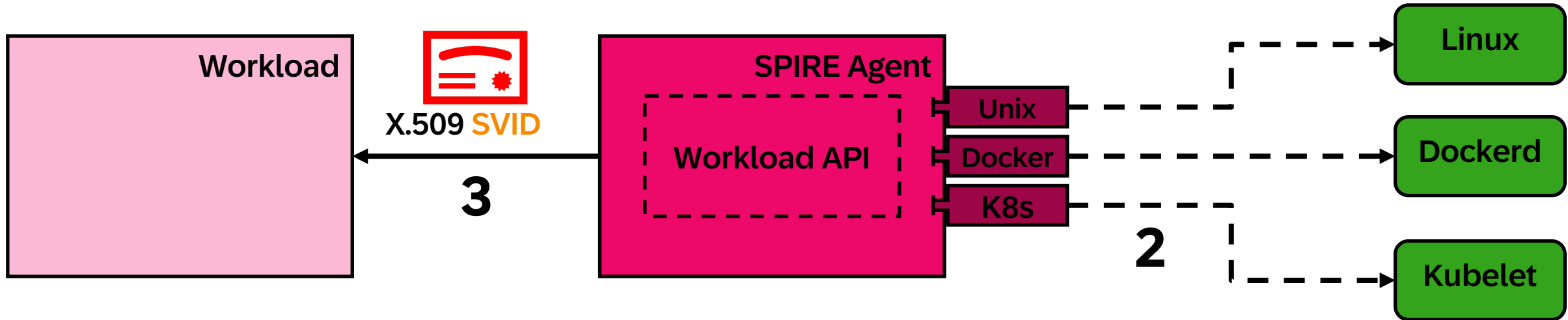
Workload Attestation



Workload Attestation

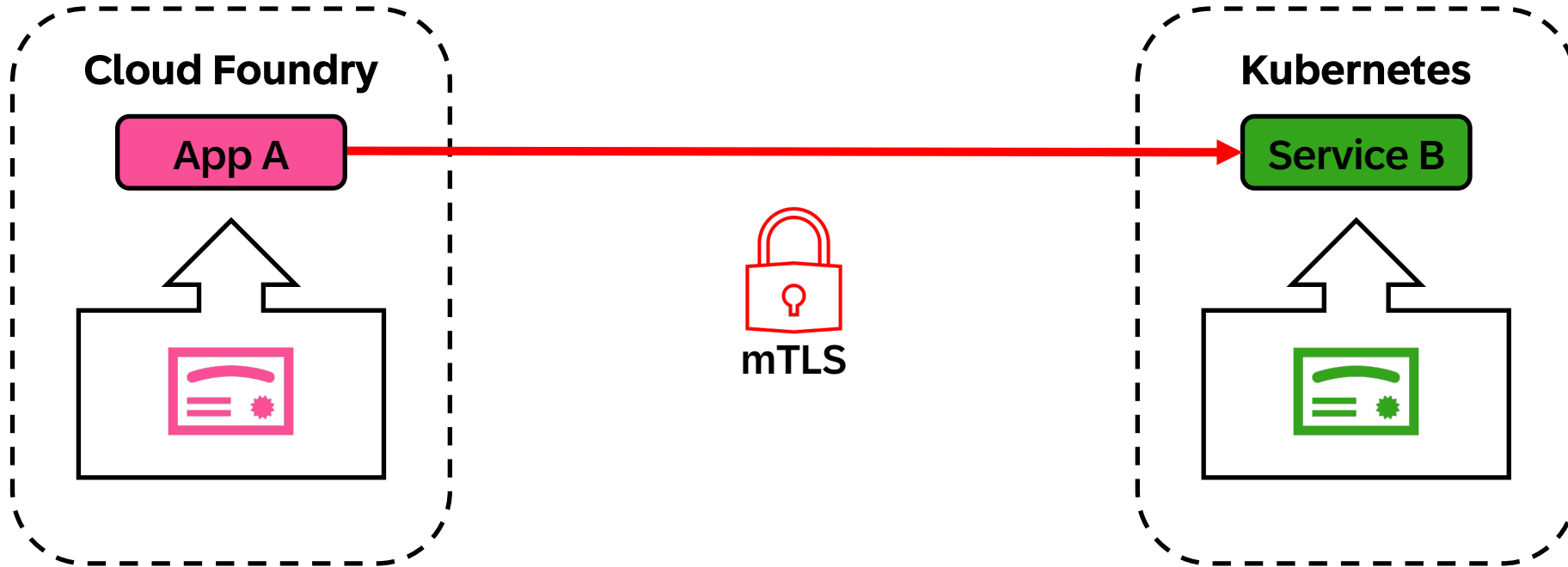


Workload Attestation



Use case

Setup Today



How to get a client certificate?

1. Generate a Private Key:

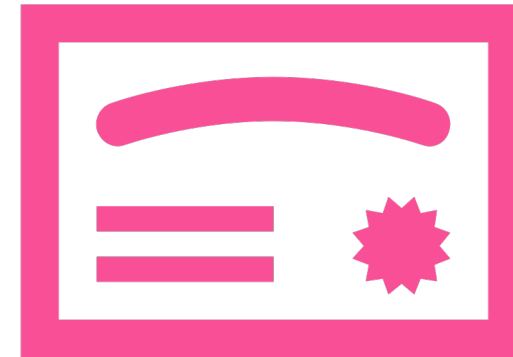
```
openssl genrsa -out client.key 2048
```

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": "..."
```



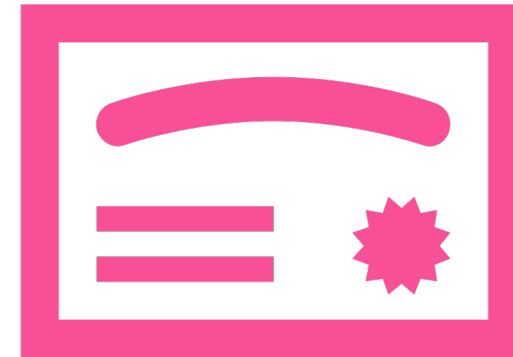
How to get a client certificate?

4. CA signs the CSR and returns the chain

```
extract_client_certificate_from_chain(cert_chain)
```

5. Read the signed certificate

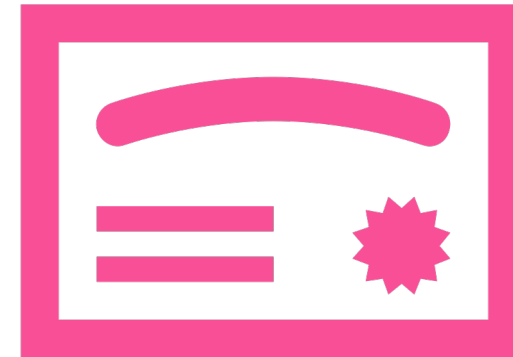
```
openssl x509 -text -noout -in client.pem
```



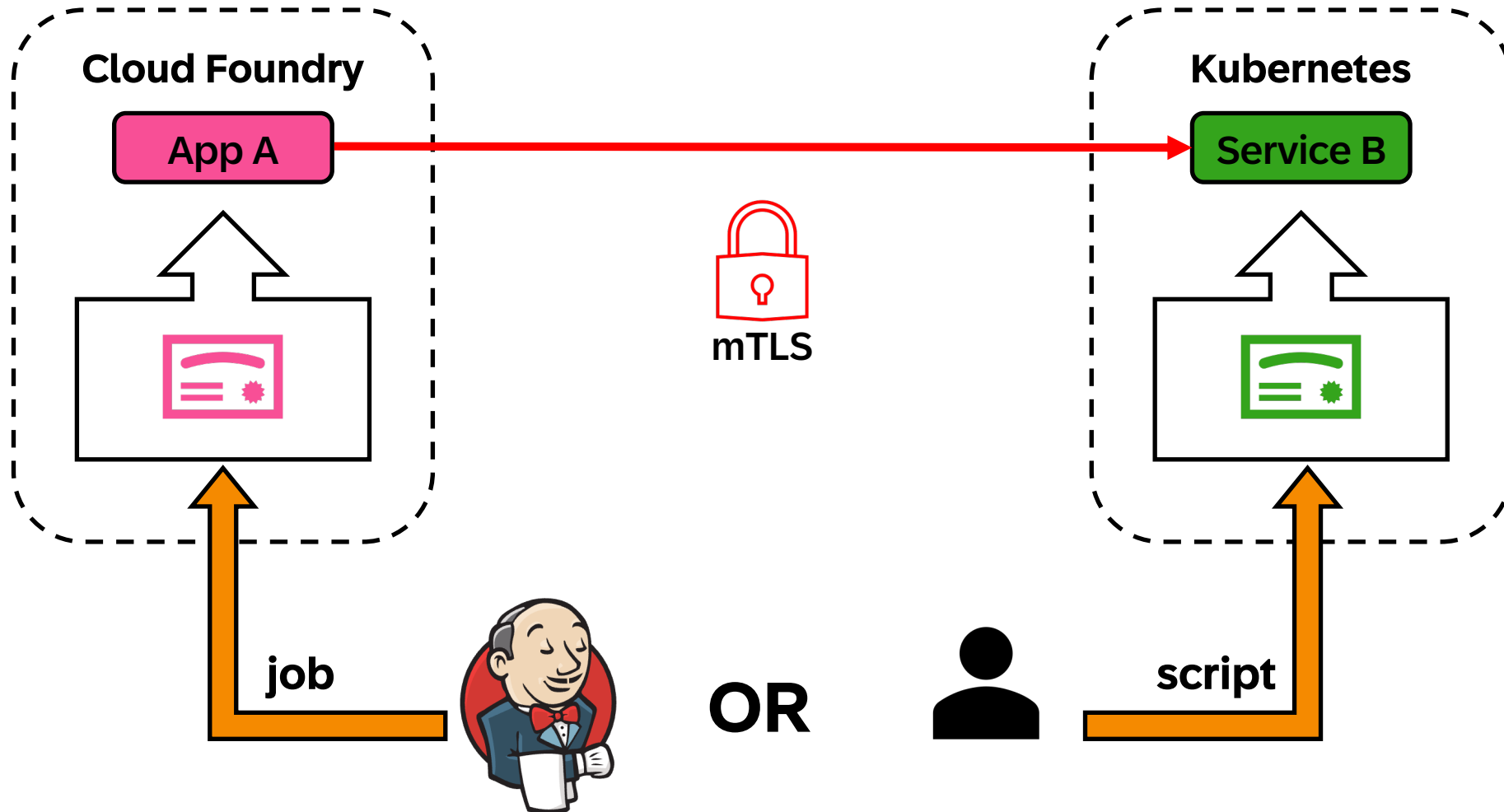
How to get a client certificate?

In a Nutshell

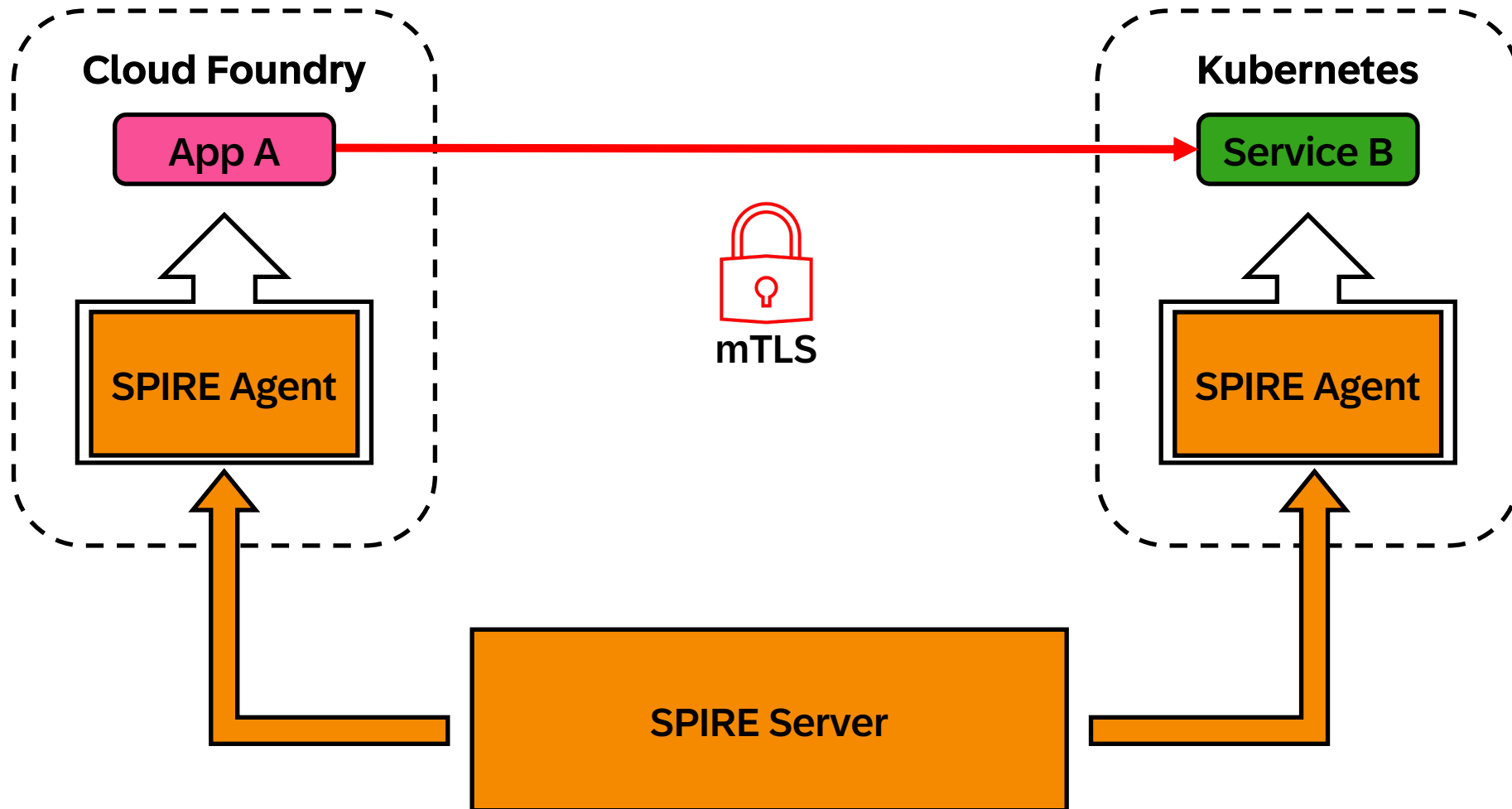
- **Mutli-Step Process**
- **Secrets are Involved**
- **Many Decisions with Impact on Security**



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

