KubeCon | CloudNativeCon

North America 2023

Leveraging SPIRE for All Your Production Identity Needs

Chirag Kapadia
Senior Software Engineer, Uber

Ryan Turner
Staff Software Engineer, Uber; SPIRE Maintainer

Agenda

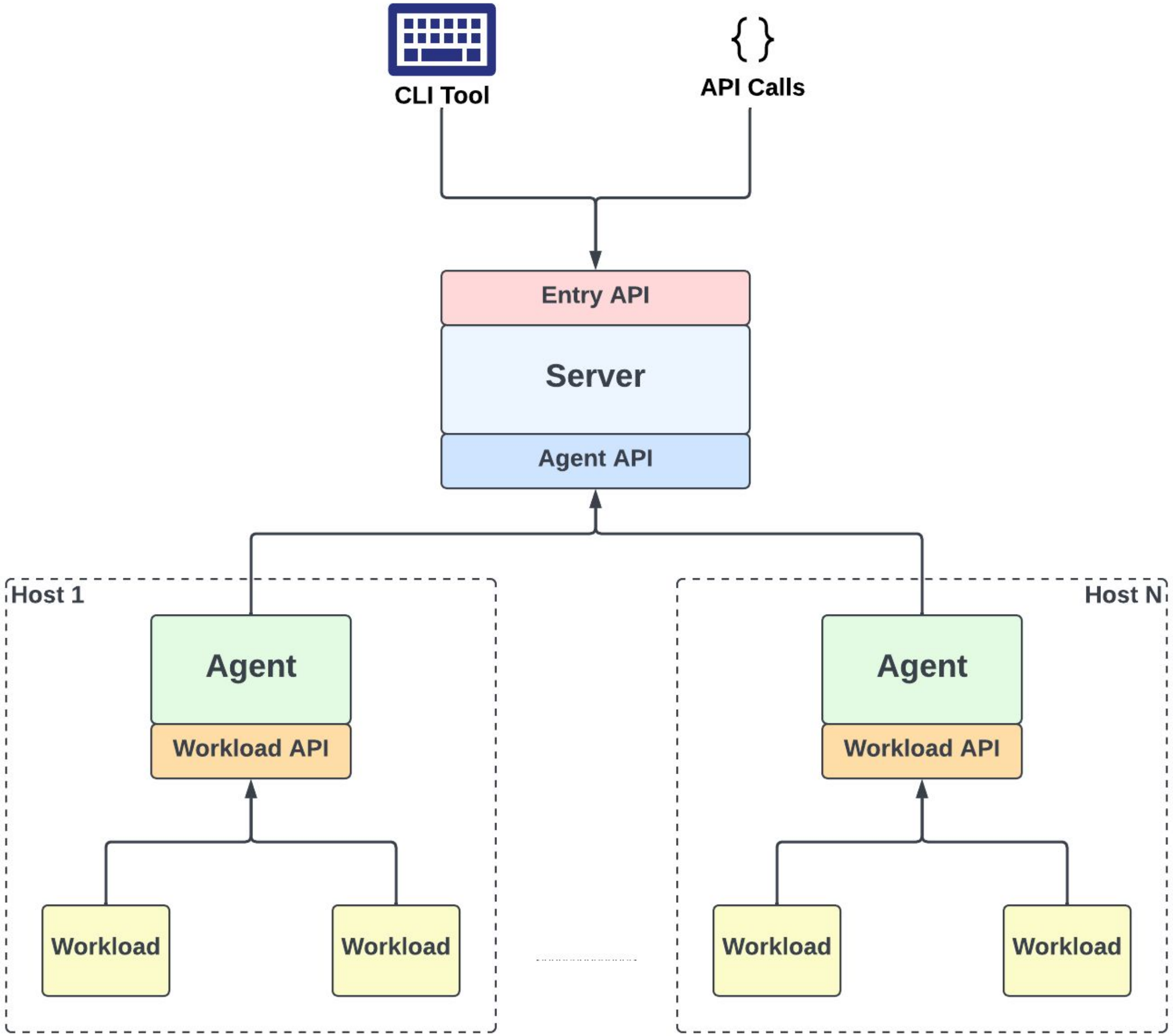
- SPIFFE Overview
- SPIRE Overview and Architecture
- SPIRE Plugins
- SPIRE issued X.509 Certificates
- CredentialComposer Plugin
- Use-case: Encrypted MySQL connections
- Demo!
- QnA

SPIFFE - Secure Production Identity Framework for Everyone

- SPIFFE ID - URI string representing “name” of an entity
  - Format - `spiffe://trust-domain/path`
    - Trust Domain - trust root of the system
    - Path - unique for every workload
- SVID: SPIFFE Verifiable Identity Document
  - Document carrying the SPIFFE ID, identifying the presenter
  - Supports X.509 and JWT document types
- Workload API - Specification for workloads to obtain SVIDs

SPIRE - SPIFFE Runtime Environment

- Production-ready implementation of the SPIFFE APIs
- Performs node and workload attestation to securely distribute SVIDs to workloads
- Workloads can use SVIDs to authenticate to other services
  - mTLS encryption using X.509-SVIDs.
  - Bearer token based authentication using JWT-SVIDs.


CLI Tool
{ }
API Calls
Entry API
Server
Agent API
Host 1
Agent
Workload API
Workload
Workload
Host N
Agent
Workload API
Workload
Workload

SPIRE Plugins

SPIRE is highly extensible via a plugin framework that allows many core operations to be added and customized.

Server Plugin Types

- KeyManager
- NodeAttestor
- UpstreamAuthority
- BundlePublisher / Notifier
- CredentialComposer

Agent Plugin Types

- KeyManager
- NodeAttestor
- WorkloadAttestor
- SVIDStore

X.509 SVIDs

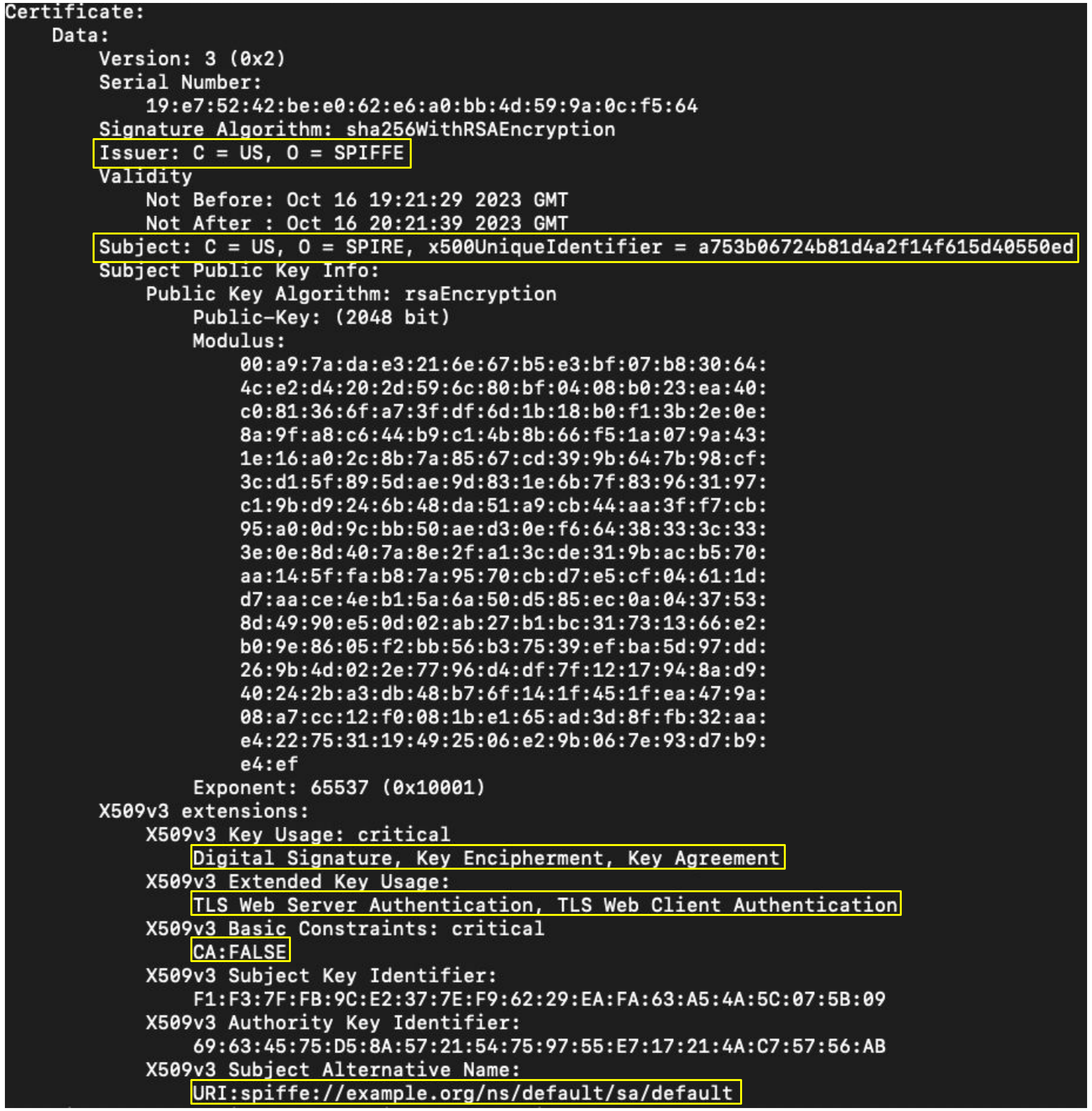
Encoding of SVID information into an X.509 Certificate.

- Contains SPIFFE ID as URI SAN Extension in the X.509 Certificate.
- Leaf Certificates -
  - Used for identifying a workload in a trust domain.
- Signing Certificates -
  - SPIRE Server CA certificate, used for signing Leaf certificates.
  - Authority of a trust domain.

SPIRE X.509 SVIDs

Leaf Certificate Format:

- URI Subject Alternative Name (SAN)
  - SPIFFE ID of workload
- Issuer
  - `C=US, O=SPIFFE`
- Subject:
  - `C=US, O=SPIRE, x500UniqueIdentifier=<id>`
  - `id` is unique for a workload
- Key Usage
  - `Digital Signature`
  - `Key Encipherment`
  - `Key Agreement`
- Extended Key Usage
  - `TLS Web Server Authentication`
  - `TLS Web Client Authentication`
- `CA:FALSE`



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            19:e7:52:42:be:e0:62:e6:a0:bb:4d:59:9a:0c:f5:64
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = US, O = SPIFFE
        Validity
            Not Before: Oct 16 19:21:29 2023 GMT
            Not After : Oct 16 20:21:39 2023 GMT
        Subject: C = US, O = SPIRE, x500UniqueIdentifier = a753b06724b81d4a2f14f615d40550ed
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:a9:7a:da:e3:21:6e:67:b5:e3:bf:07:b8:30:64:
                    4c:e2:d4:20:2d:59:6c:80:bf:04:08:b0:23:ea:40:
                    c0:81:36:6f:a7:3f:df:6d:1b:18:b0:f1:3b:2e:0e:
                    8a:9f:a8:c6:44:b9:c1:4b:8b:66:f5:1a:07:9a:43:
                    1e:16:a0:2c:8b:7a:85:67:cd:39:9b:64:7b:98:cf:
                    3c:d1:5f:89:5d:ae:9d:83:1e:6b:7f:83:96:31:97:
                    c1:9b:d9:24:6b:48:da:51:a9:cb:44:aa:3f:f7:cb:
                    95:a0:0d:9c:bb:50:ae:d3:0e:f6:64:38:33:3c:33:
                    3e:0e:8d:40:7a:8e:2f:a1:3c:de:31:9b:ac:b5:70:
                    aa:14:5f:fa:b8:7a:95:70:cb:d7:e5:cf:04:61:1d:
                    d7:aa:ce:4e:b1:5a:6a:50:d5:85:ec:0a:04:37:53:
                    8d:49:90:e5:0d:02:ab:27:b1:bc:31:73:13:66:e2:
                    b0:9e:86:05:f2:bb:56:b3:75:39:ef:ba:5d:97:dd:
                    26:9b:4d:02:2e:77:96:d4:df:7f:12:17:94:8a:d9:
                    40:24:2b:a3:db:48:b7:6f:14:1f:45:1f:ea:47:9a:
                    08:a7:cc:12:f0:08:1b:e1:65:ad:3d:8f:fb:32:aa:
                    e4:22:75:31:19:49:25:06:e2:9b:06:7e:93:d7:b9:
                    e4:ef
                Exponent: 65537 (0x10001)
        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 Key Identifier:
                F1:F3:7F:FB:9C:E2:37:7E:F9:62:29:EA:FA:63:A5:4A:5C:07:5B:09
            X509v3 Authority Key Identifier:
                69:63:45:75:D5:8A:57:21:54:75:97:55:E7:17:21:4A:C7:57:56:AB
            X509v3 Subject Alternative Name:
                URI:spiffe://example.org/ns/default/sa/default
```

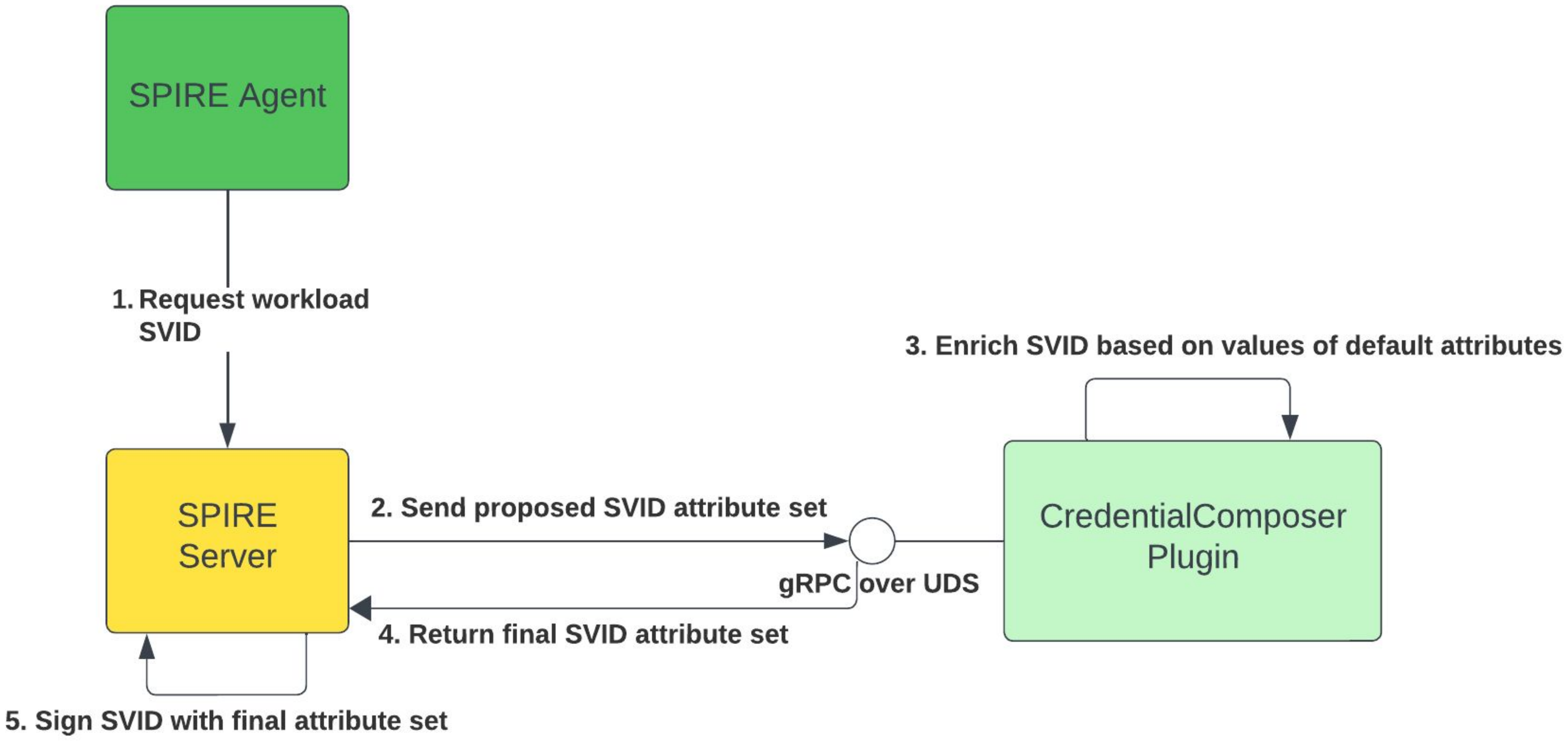
Challenges with SPIRE-Issued Certificates

- Authentication to legacy/non-SPIFFE aware systems
- X.509 extensions for PKI interoperability
  - CRL Distribution Point
  - OCSP Server
  - Code signing
  - Path validation policies
- Conformance with organization PKI practices

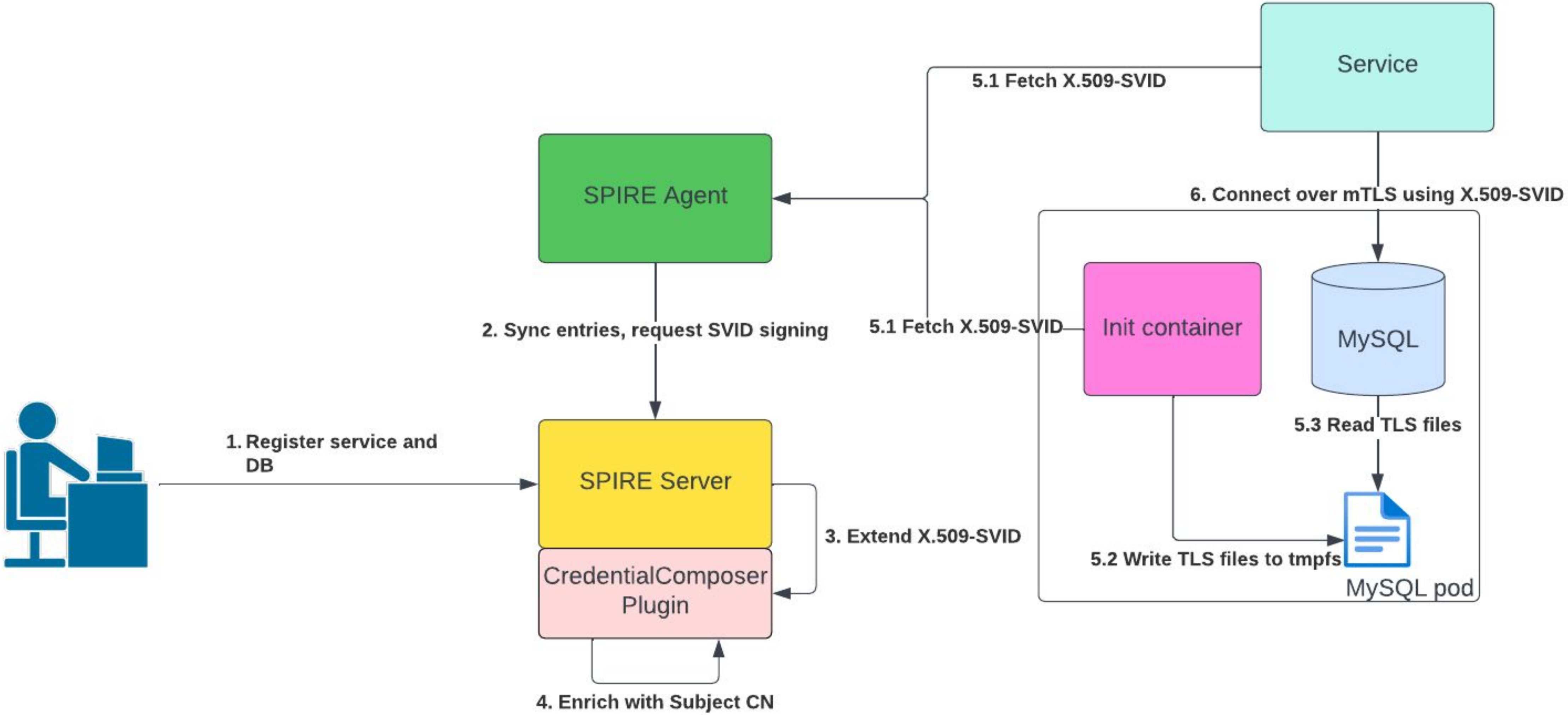
CredentialComposer Plugin

- Enables customization of SVID attributes
- X.509-SVID
    - Types of certificates that can be customized
        - Server CA
        - Server TLS server certificate
        - Workload leaf
        - Agent
    - Fields that can be customized
        - Subject
        - DNS SANs
        - Extensions
- JWT-SVID
        - Arbitrary JWT claims (excluding `sub`)

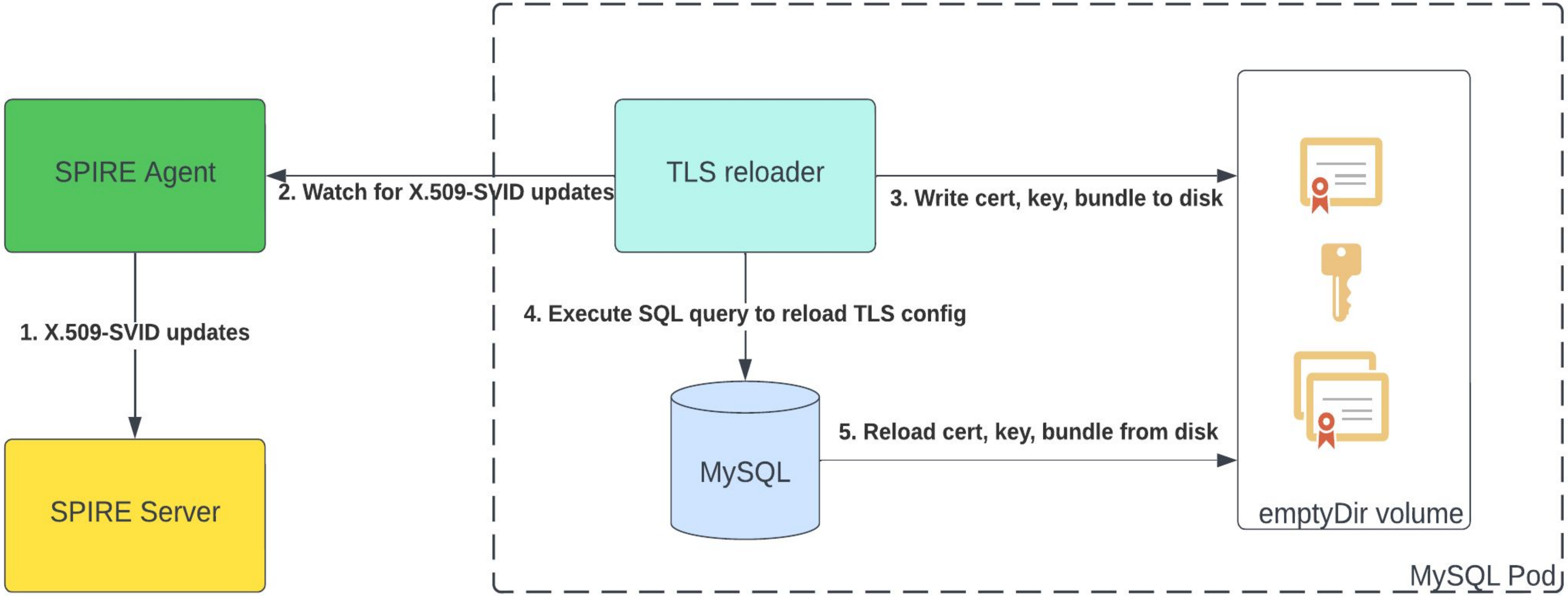
CredentialComposer Plugin


SPIRE Agent
1. Request workload SVID
SPIRE Server
2. Send proposed SVID attribute set
gRPC over UDS
4. Return final SVID attribute set
5. Sign SVID with final attribute set
3. Enrich SVID based on values of default attributes
CredentialComposer Plugin

Demo


Service
5.1 Fetch X.509-SVID
SPIRE Agent
6. Connect over mTLS using X.509-SVID
2. Sync entries, request SVID signing
5.1 Fetch X.509-SVID
Init container
MySQL
5.3 Read TLS files
1. Register service and DB
SPIRE Server
3. Extend X.509-SVID
CredentialComposer Plugin
5.2 Write TLS files to tmpfs
MySQL pod
4. Enrich with Subject CN

Demo - Rotation


SPIRE Agent
2. Watch for X.509-SVID updates
TLS reloader
3. Write cert, key, bundle to disk
1. X.509-SVID updates
4. Execute SQL query to reload TLS config
MySQL
5. Reload cert, key, bundle from disk
SPIRE Server
emptyDir volume
MySQL Pod

Please scan the QR Code above
to leave feedback on this session

Appendix

- Demo source code: https://github.com/rturner3/spire-mysql-demo
- CredentialComposer plugin interface: https://github.com/spiffe/spire-plugin-sdk/blob/main/proto/spire/plugin/server/credentialcomposer/v1/credentialcomposer.proto
- CredentialComposer GitHub issue: https://github.com/spiffe/spire/issues/3253
- SPIFFE Slack: https://slack.spiffe.io
- SPIRE GitHub: https://github.com/spiffe/spire