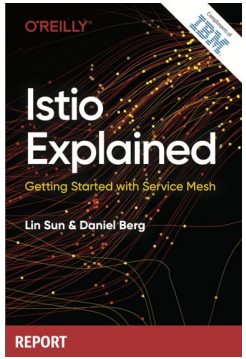
External CA Integration with Istio Explained

Lin Sun & Josh van Leeuwen

Lin Sun



6500+ contributions

TOC & Steering Member

Director of Open Source, Solo.io

@linsun_unc

lin.sun@solo.io

linkedin.com/pub/lin-sun/1/…

Ambassador

Josh van Leeuwen

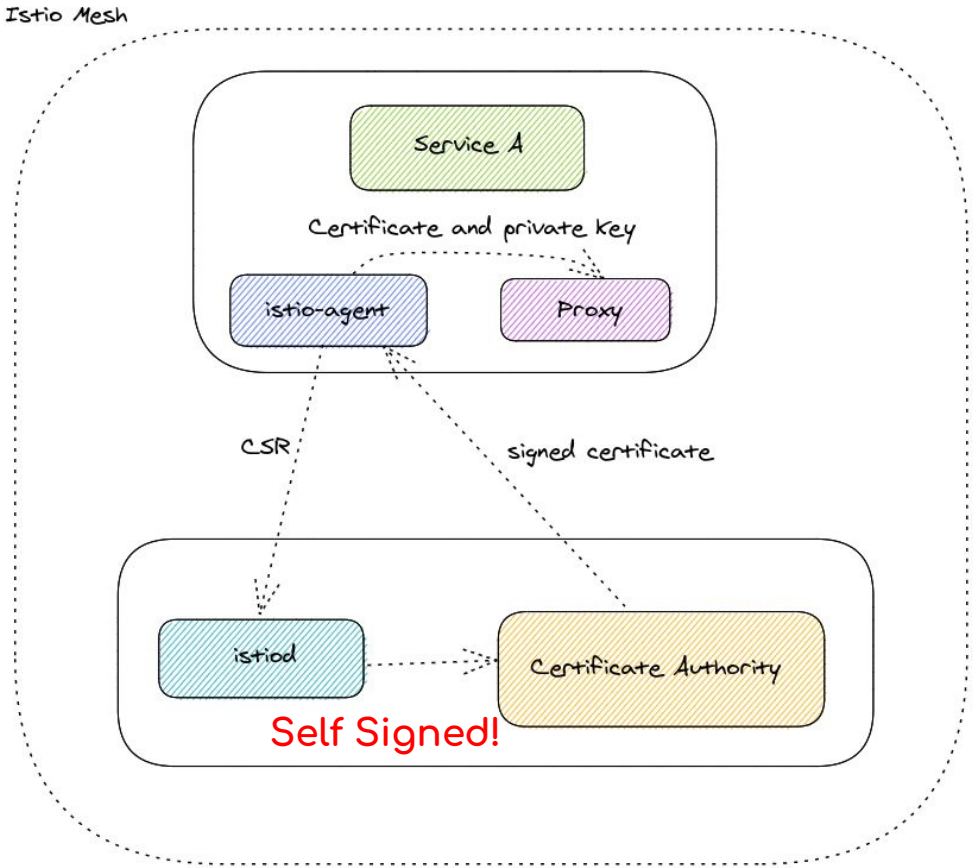
@joshvanl

joshua.vanleeuwen@jetstack.io

linkedin.com/in/joshvanl

Staff Software Engineer

Istio Defaults

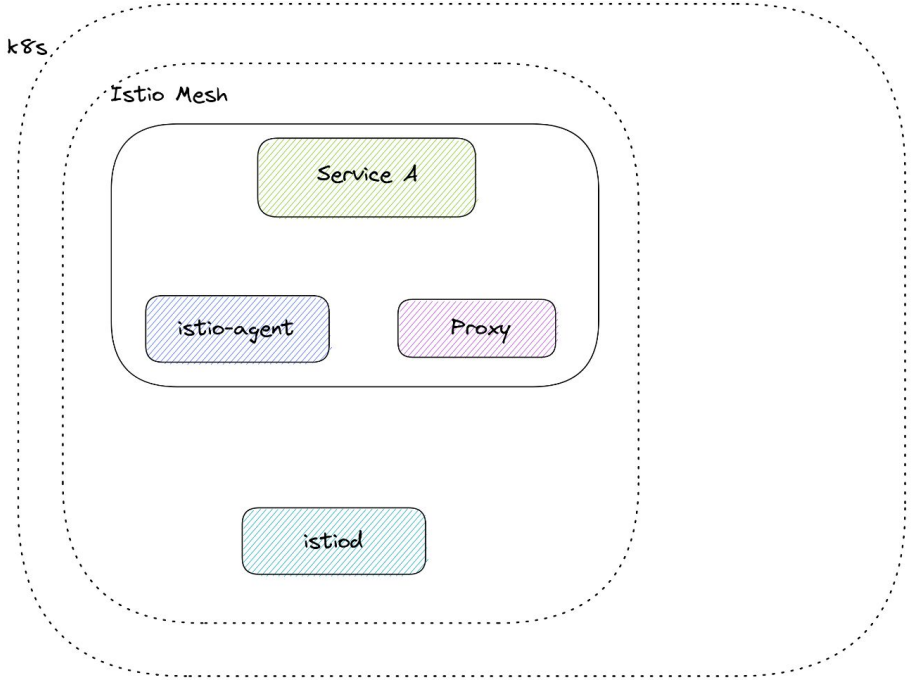


Self-Signed Root CA

```
~/ kubectl get secret -n istio-system
NAME                                          TYPE                                  DATA   AGE
default-token-jpq2n                           kubernetes.io/service-account-token   3      48m
istio-ca-secret                               istio.io/ca-root                      5      48m
istio-ingressgateway-service-account-token-brmfm   kubernetes.io/service-account-token   3      48m
istio-reader-service-account-token-n6xbx      kubernetes.io/service-account-token   3      48m
istiod-service-account-token-6ffm6            kubernetes.io/service-account-token   3      48m
istiod-token-6q2mn                            kubernetes.io/service-account-token   3      48m
~/ kubectl get cm -n istio-system
NAME                                   DATA   AGE
istio                                  2      48m
istio-ca-root-cert                     1      48m
istio-gateway-deployment-leader        0      48m
istio-gateway-status-leader            0      48m
istio-leader                           0      48m
istio-namespace-controller-election    0      48m
istio-sidecar-injector                 2      48m
kube-root-ca.crt                       1      48m
```

```
~/ kubectl get cm istio-ca-root-cert -o json | jq '[.data[]][0]' -r | openssl x509 -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            e8:1f:61:b2:8b:e1:8b:50:b1:1b:57:0d:c2:61:68:85
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = cluster.local
        Validity
            Not Before: Apr 21 01:15:24 2022 GMT
            Not After : Apr 18 01:15:24 2032 GMT
        Subject: O = cluster.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:bd:f0:cf:7c:f5:45:04:61:18:03:86:ce:6e:90:
                    3c:a2:fa:cc:53:fd:4c:6c:d4:db:6a:1b:97:13:1a:
                    b6:bf:0d:c9:da:fe:c1:e3:6e:61:18:97:5f:89:c0:
                    d2:62:00:34:03:1b:81:e8:80:5a:f1:a6:22:85:44:
                    59:8c:43:28:85:7f:22:8d:62:8c:80:99:70:a4:de:
                    a6:0f:d8:ed:a6:6a:33:95:9d:f3:a8:7c:04:b9:cc:
                    85:d6:16:a3:b3:be:d5:70:a1:3b:57:98:a4:e0:19:
                    03:57:57:86:63:4c:8a:4c:20:2f:39:ff:05:7d:33:
                    a9:85:c4:fb:38:b5:95:e8:e6:4f:af:7f:9d:a7:46:
                    3e:0c:0a:33:a0:d3:71:00:9d:88:75:67:ed:c4:96:
                    9e:ed:c3:eb:b2:b6:1d:b6:9d:35:8d:bd:25:e5:32:
                    14:9e:a9:24:d2:12:88:15:52:03:b3:ab:c7:d6:87:
                    40:ff:1c:b5:e2:14:7c:74:3d:52:18:c8:b7:97:1f:
                    cf:93:85:db:8e:d2:1c:cb:1b:6e:92:9e:55:cc:1d:
                    25:bb:71:e5:40:ef:eb:a6:56:e1:27:e3:c9:f1:8d:
                    fc:9b:a3:61:aa:c5:a2:72:0d:5f:26:32:e3:c7:f9:
                    1d:09:2a:4e:b6:2b:b6:e4:58:c1:ee:fb:05:26:f9:
                    2d:db
                Exponent: 65537 (0x10001)
```

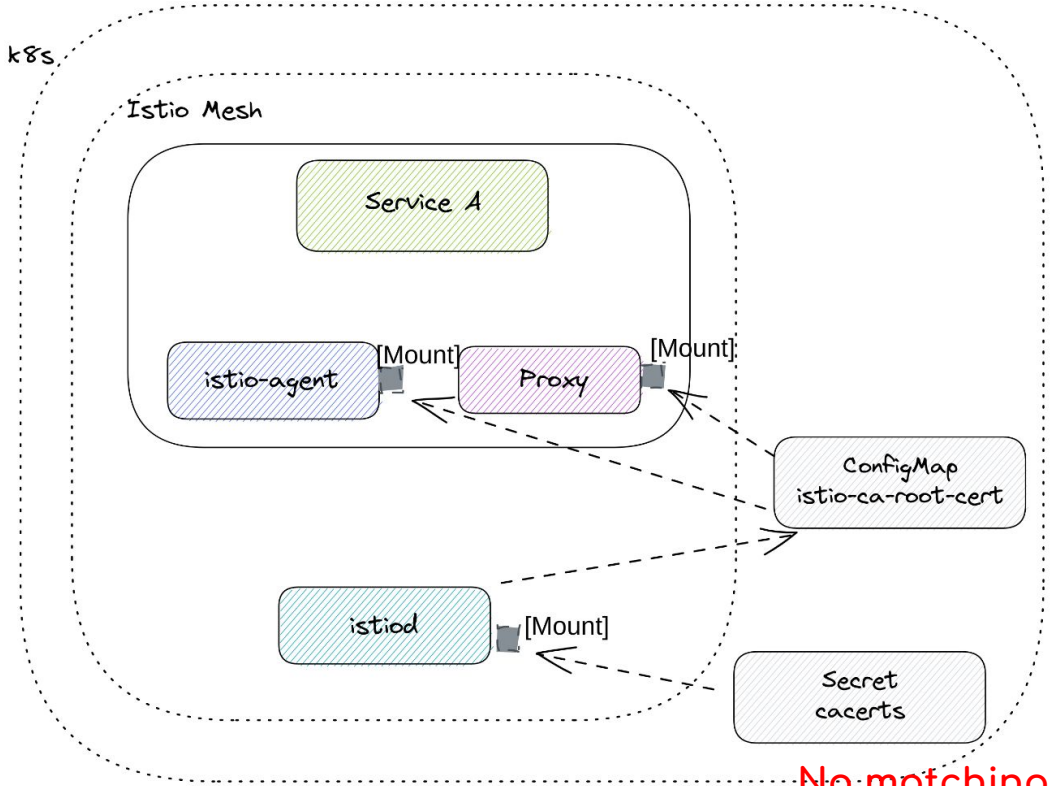

Trust Distribution



Trust Distribution

```
~/ kubectl get ns
NAME                 STATUS   AGE
default              Active   52m
istio-system         Active   51m
kube-node-lease      Active   52m
kube-public          Active   52m
kube-system          Active   52m
local-path-storage   Active   52m
~/ kubectl get cm -A -l istio.io/config=true
NAMESPACE      NAME                 DATA   AGE
default        istio-ca-root-cert   1      52m
istio-system   istio-ca-root-cert   1      52m
~/ k create ns foo
namespace/foo created
~/ kubectl get ns
~/ kubectl get cm -A -l istio.io/config=true
NAMESPACE      NAME                 DATA   AGE
default        istio-ca-root-cert   1      52m
foo            istio-ca-root-cert   1      17s
istio-system   istio-ca-root-cert   1      52m
```


Trust Distribution



Trust Distribution

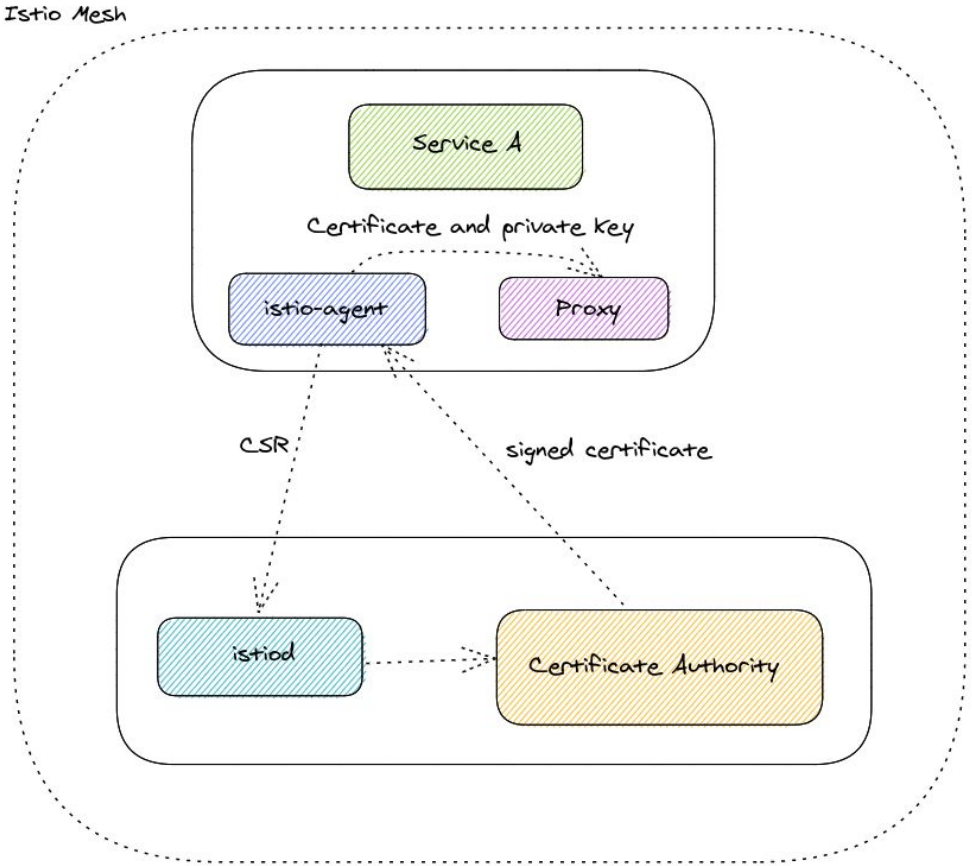
Istio-agent sends CSR request to Istiod to get private key signed (explain trust distribution)

- Workload Private key never leave the machine, not persisted either.

Workload cert expires in 24 hours

Workload cert rotates in 12 hours

- Configurable via pilot-agent env var: SECRET_GRACE_PERIOD_RATIO (default 0.5)



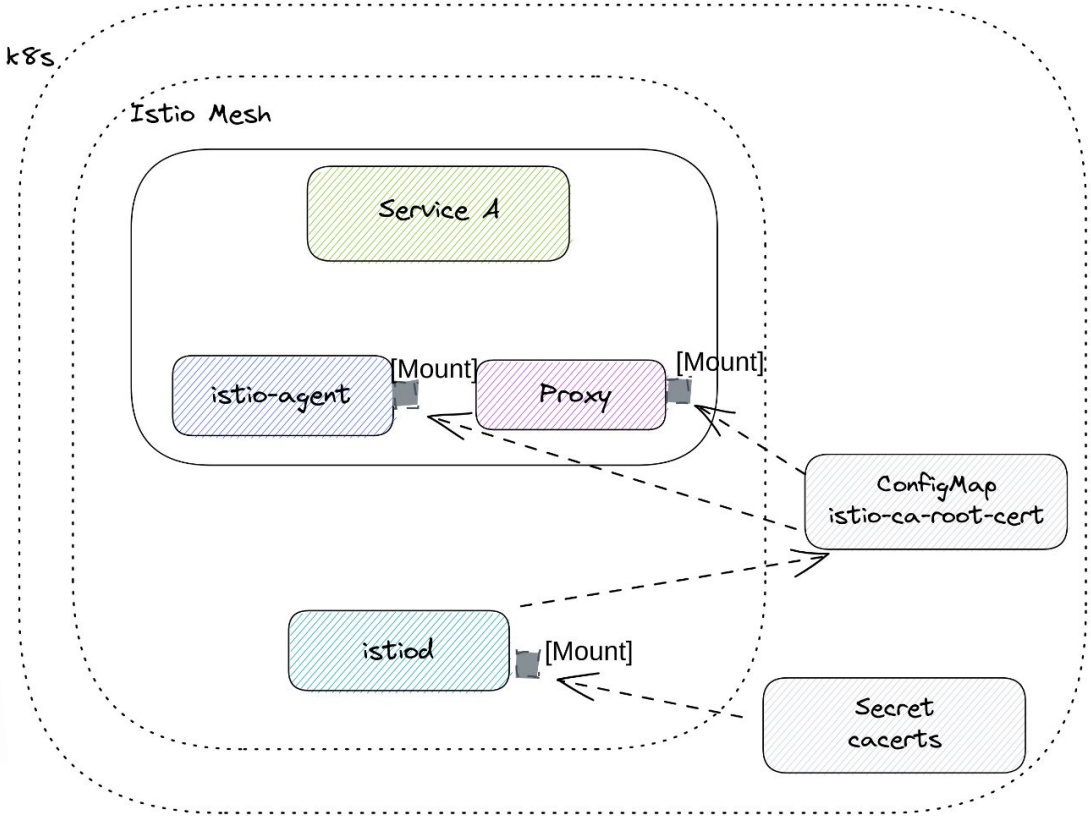
Plugin your own certificates for Istio CA

- Cacerts
- Configure it via IstioOperator API (meshConfig.caCertificates)

With cacerts

- Populate cacets Secret in istio-system, ingested by istiod
  - ca-cert.pem: the generated intermediate certificates
  - ca-key.pem: the generated intermediate key
  - cert-chain.pem: the generated certificate chain which is used by istiod
  - root-cert.pem: the root certificate

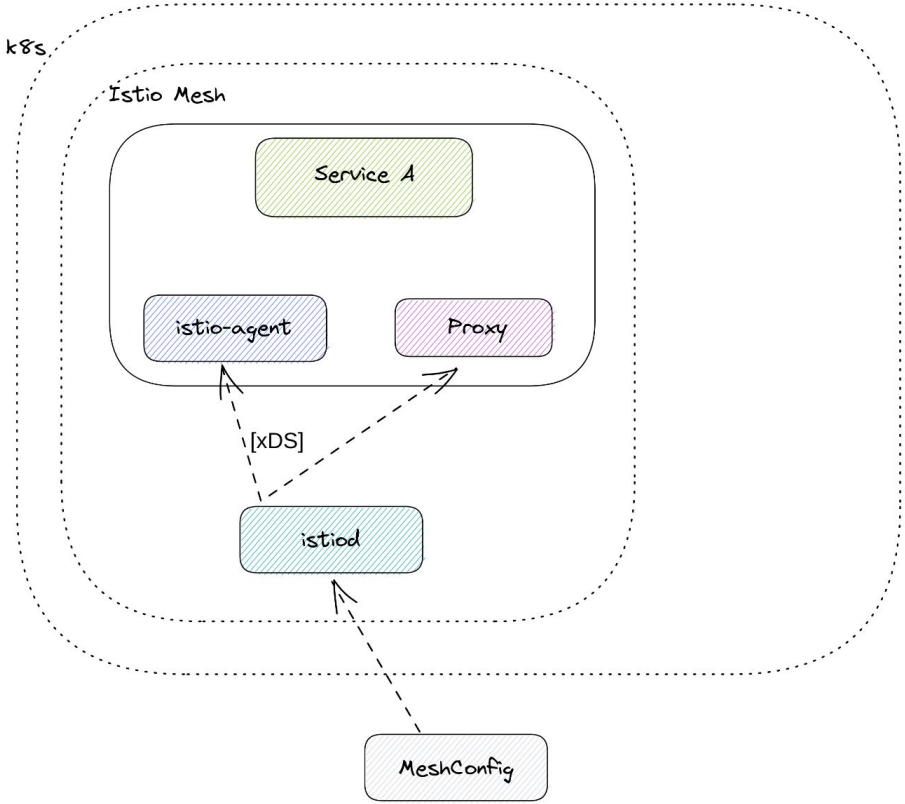
- Behaves the same as default, but with bring your own intermediate



Add support to istiod to notice cacerts changes #31522

Merged istio-testing merged 9 commits into istio:master from rveerama1:istiod-restart on Jul 20, 2021

With MeshConfig



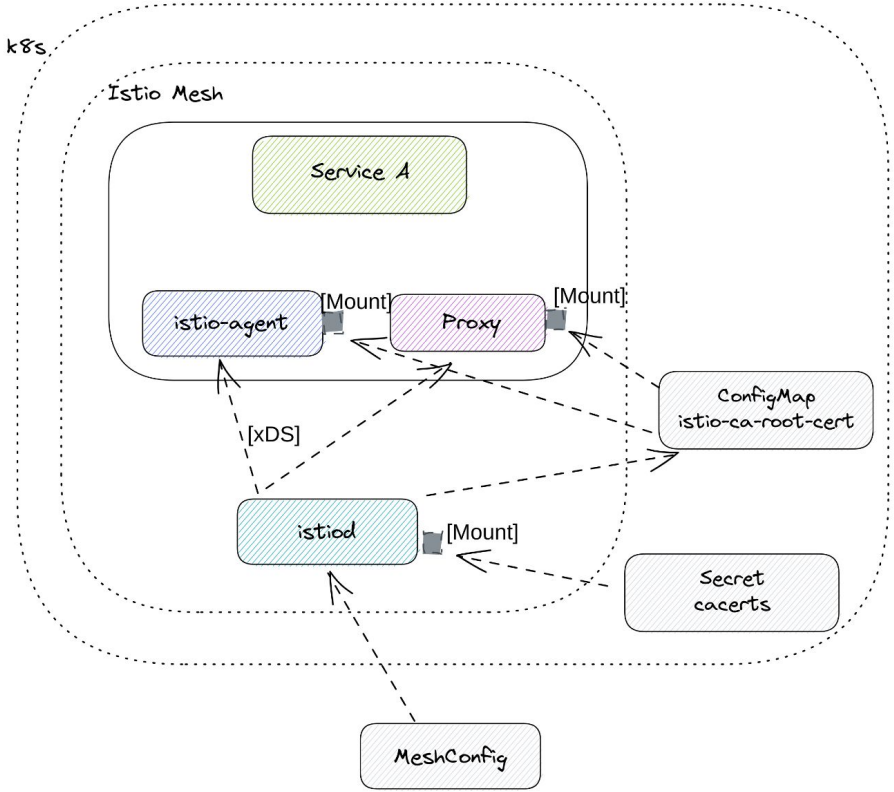
- Global Mesh Config Trust Anchors
- Pushed out of all workloads by Istiod + Istio agent
- Compatible with spiffe compliant servers such as spire

```yaml
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  namespace: istio-system
spec:
  profile: "demo"
  hub: gcr.io/istio-release
  meshConfig:
    trustDomain: foo.bar
    caCertificates:
    - spiffeBundleUrl: "https://spire.spiffe"
    - pem: |
        -----BEGIN CERTIFICATE-----
        MIIDTDCCAjSgAwIBAgIQUDRiWJ9L2NGL3Ibxa6dk/TANBgkqhkiG9w0BAQsFADBA
        MSswEwYDVQQKEwxjZXJ0LW1hbmFnZXIwFAYDVQQKEw1jbHVzdGVyLmxvY2FsMREw
        DwYDVQQDEwhpc3Rpby1jYTAeFw0yMTA4MDMxMDM2MjBaFw0yMTExMDExMDM2MjBa
        MEAxKzATBgNVBAoTDGNlcnQtbWFuYWdlcjAUBgNVBAoTDWNsdXN0ZXIubG9jYWwx
        ETAPBgNVBAMTCGlzdGlvLWNhMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
        AQEAuM5xUppoXwoeWWGFhiviwkY5rDxriW5bt9ZO4gYf7/NiBjC68szMxls+XTn3
        SPhx7u0FcBaI7C0a6rYNJxflNuUQ5E+BBjuUd0tnxU6UGIIERPb5ZDtsxZQJq9qh
        eYkRgiDEAi5MKp0+O0uAh9JCFOSq3aZMVS9lQ5ib/S4gNzIiC5skpSVIBrXhtEZV
        foi5VmcTOfS/kDR0avh+yJtxh5vQtUTLf7MpuUBNrhylHF+Q6+M/pD+SpoQwe221
        KHfDLLxAeslHomJBtd2X40lx7JEv9CydjNBxw3eYN1SZ+SMxiJTBT6/KpzSbDOSY
        d7MeH8g3S0kZj09bzhRjlEokBwIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAqQwDwYD
        VR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUAmTumk1Xkr0eszilLALtnAcGGnMwDQYJ
        KoZIhvcNAQELBQADggEBAC7BS7T9tx337XB0dNsUSL67apumNrfapEzHjxXxPg5F
        dO5JFv7SqJFAh0m90yjaGL3uUKoEb+UCD+7u3J4QMbwCcIL5+i1WmFgg5Zan6FN5
        7W7ImlHqH43Rxw/q6qBrAptQfseLkFQDbRhz6lSqKT7cMUATnik546Fpt2FSle0/
        7+hYZafv+GMeUOageyZAfNVMWBvHH21G1+GNtQWdPf8od3L8QMHTAUoV5xyChR5L
        1GjLfkh4gnhyZI1kR7kVMCTQZ2GR+RLwv9X+Rjai44Sidk7GTeRicwgu9eKKX1oO
        XhihGpVDl38hJCHm2LBwEfrpH6s2Qk1ps4mqlZdmHeU=
        -----END CERTIFICATE-----
```

Can you plugin multiple CA certs for Istiod?

caCertificates	CertificateData[ ]	The extra root certificates for workload-to-workload communication. The plugin certificates (the 'cacerts' secret) or self-signed certificates (the 'istio-ca-secret' secret) are automatically added by Istiod. The CA certificate that signs the workload certificates is automatically added by Istio Agent.	No

cacerts & MeshConfig co-exist

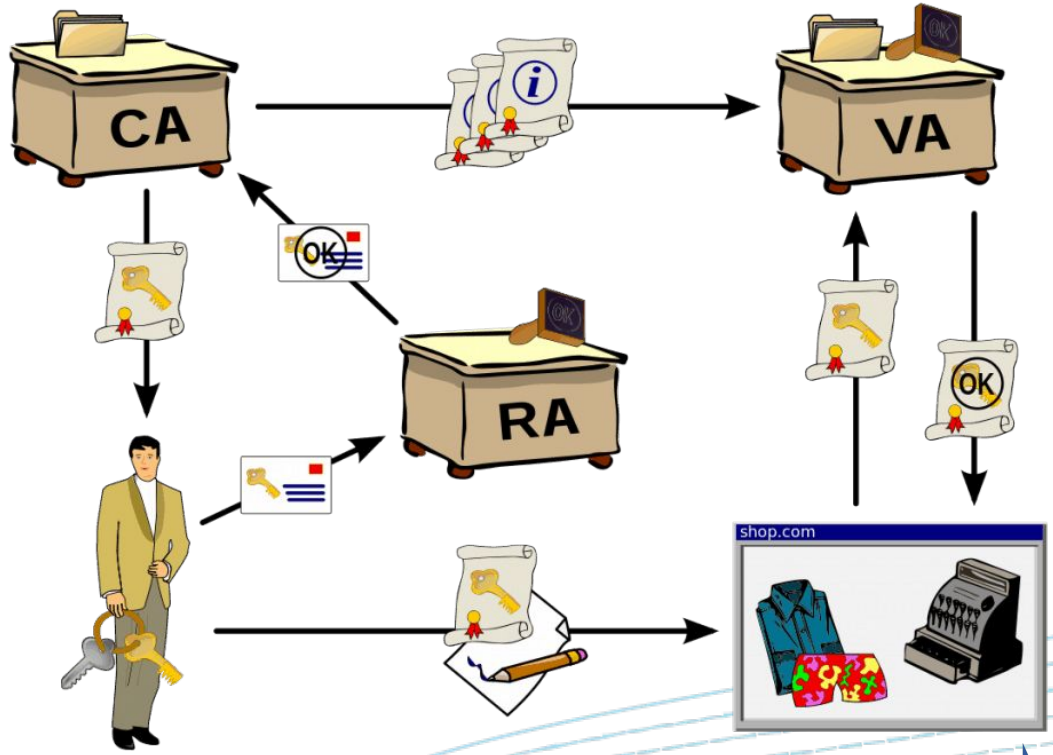


RA and CA

Registration Authority: approve requests

Certificate Authority: sign workload certs

Istiod works as RA & CA by default.

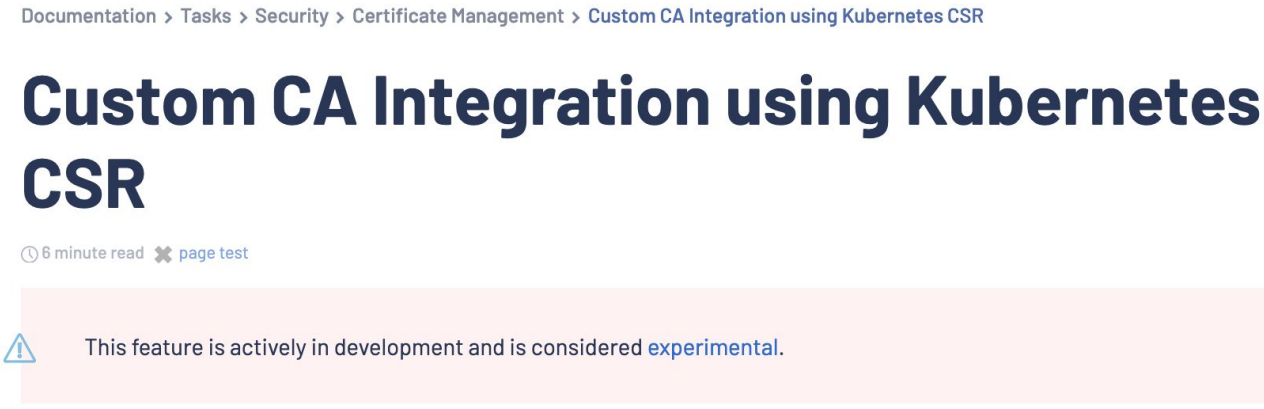


Kubernetes CSR Integration

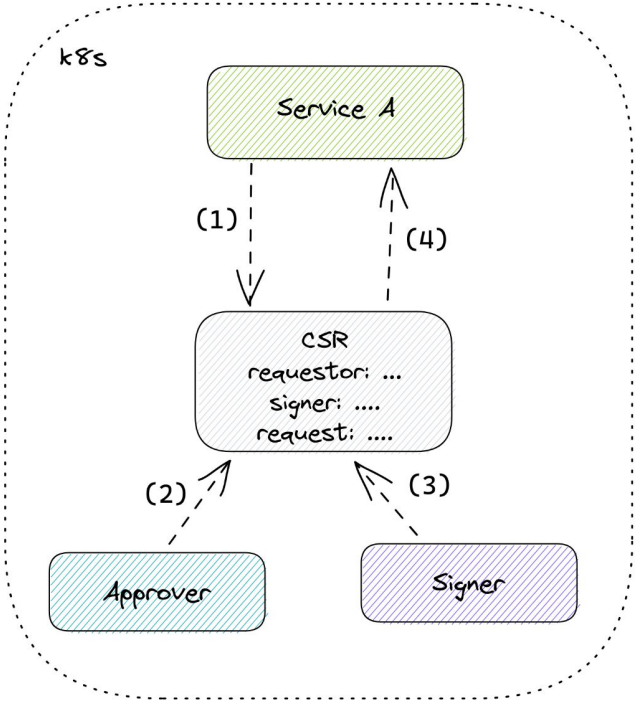
Istiod works as RA

K8s CA or other custom CAs acts as CA

Private key not present in K8s cluster

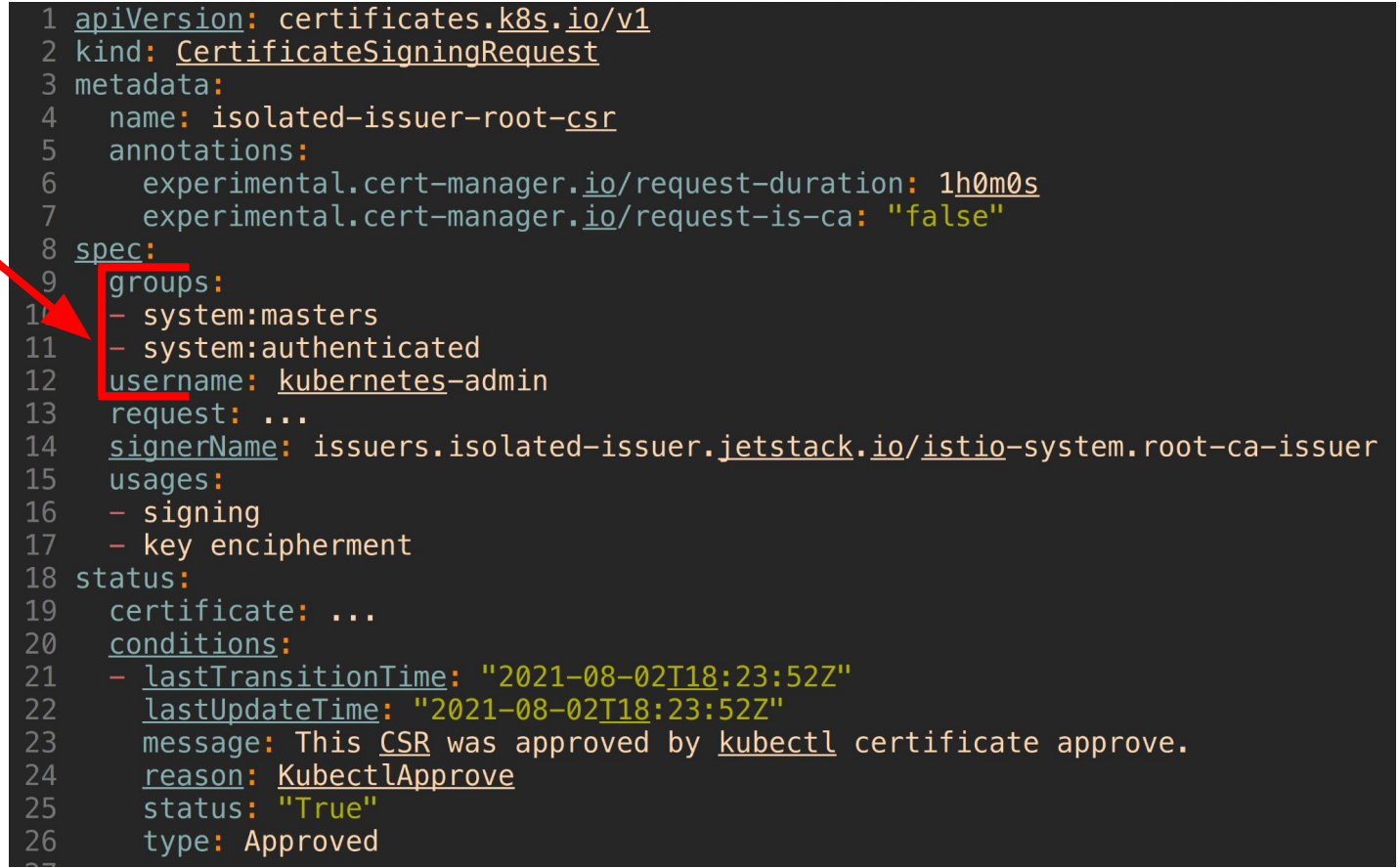


Kubernetes CSR



1. CSR requested by Service A (requestor)
2. Approver approves the request based on contents and requestor
3. Signer Signs the Approved request
4. Signed certificate read by Service A

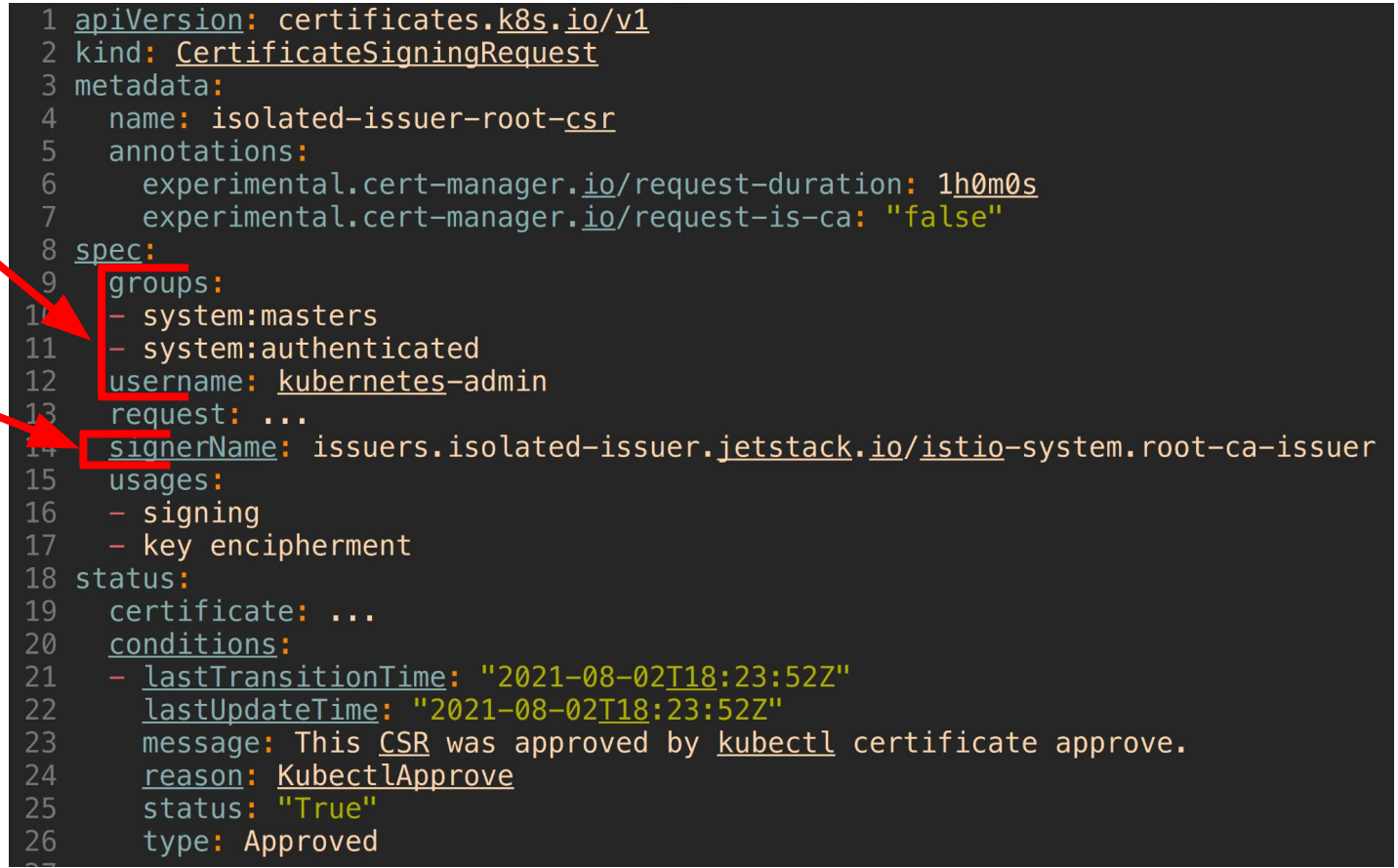
1. Requestor



```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: isolated-issuer-root-csr
  annotations:
    experimental.cert-manager.io/request-duration: 1h0m0s
    experimental.cert-manager.io/request-is-ca: "false"
spec:
  groups:
  - system:masters
  - system:authenticated
  username: kubernetes-admin
  request: ...
  signerName: issuers.isolated-issuer.jetstack.io/istio-system.root-ca-issuer
  usages:
  - signing
  - key encipherment
status:
  certificate: ...
  conditions:
  - lastTransitionTime: "2021-08-02T18:23:52Z"
    lastUpdateTime: "2021-08-02T18:23:52Z"
    message: This CSR was approved by kubectl certificate approve.
    reason: KubectlApprove
    status: "True"
    type: Approved
```

1. Requestor

2. Signer

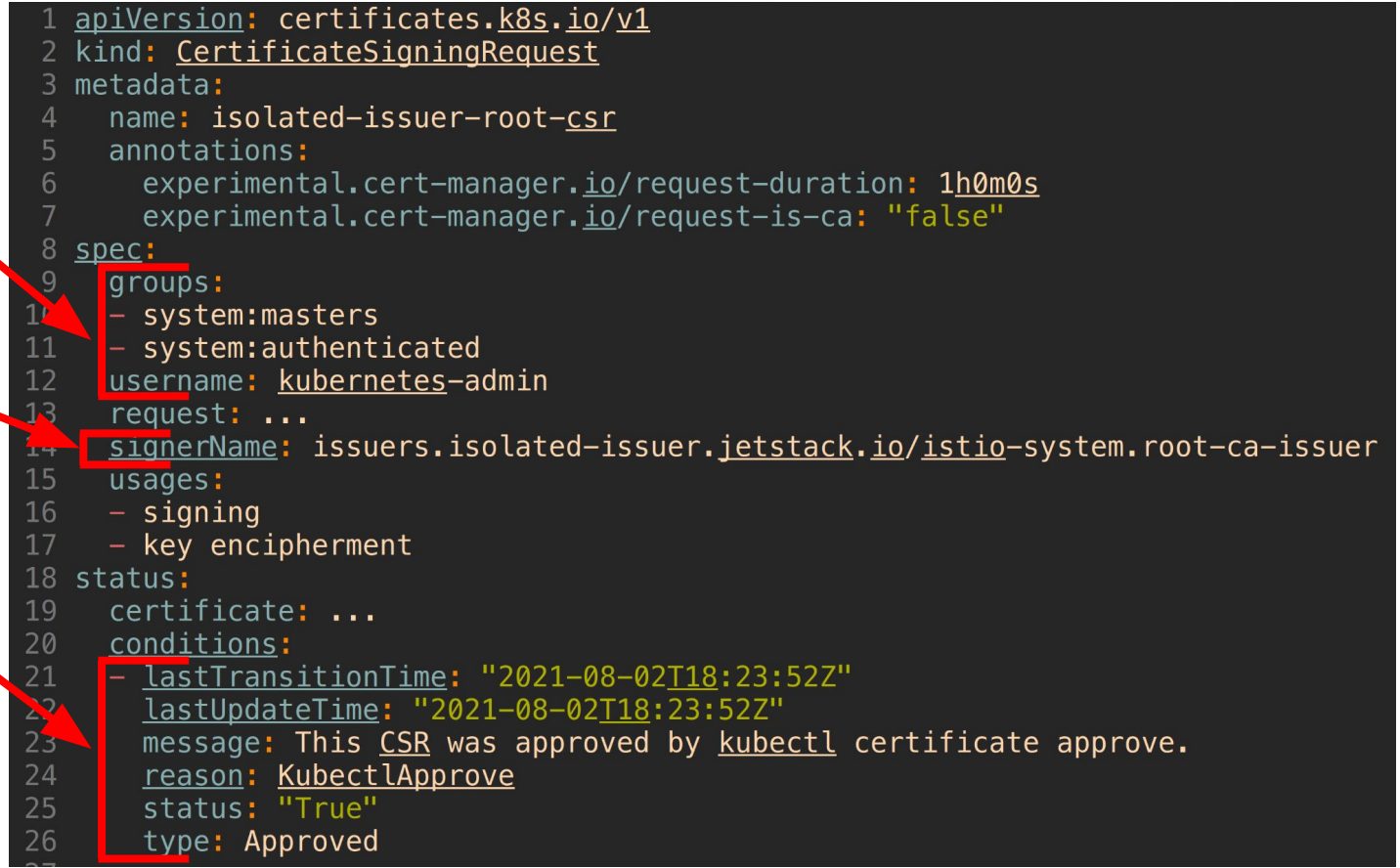


```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: isolated-issuer-root-csr
  annotations:
    experimental.cert-manager.io/request-duration: 1h0m0s
    experimental.cert-manager.io/request-is-ca: "false"
spec:
  groups:
  - system:masters
  - system:authenticated
  username: kubernetes-admin
  request: ...
  signerName: issuers.isolated-issuer.jetstack.io/istio-system.root-ca-issuer
  usages:
  - signing
  - key encipherment
status:
  certificate: ...
  conditions:
  - lastTransitionTime: "2021-08-02T18:23:52Z"
    lastUpdateTime: "2021-08-02T18:23:52Z"
    message: This CSR was approved by kubectl certificate approve.
    reason: KubectlApprove
    status: "True"
    type: Approved
```

1. Requestor

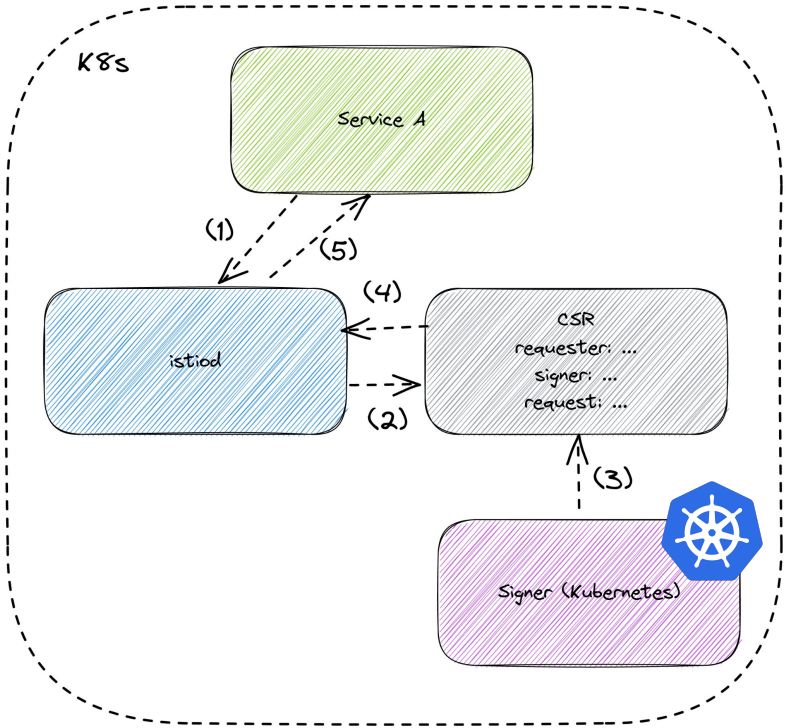
2. Signer

3. Approver



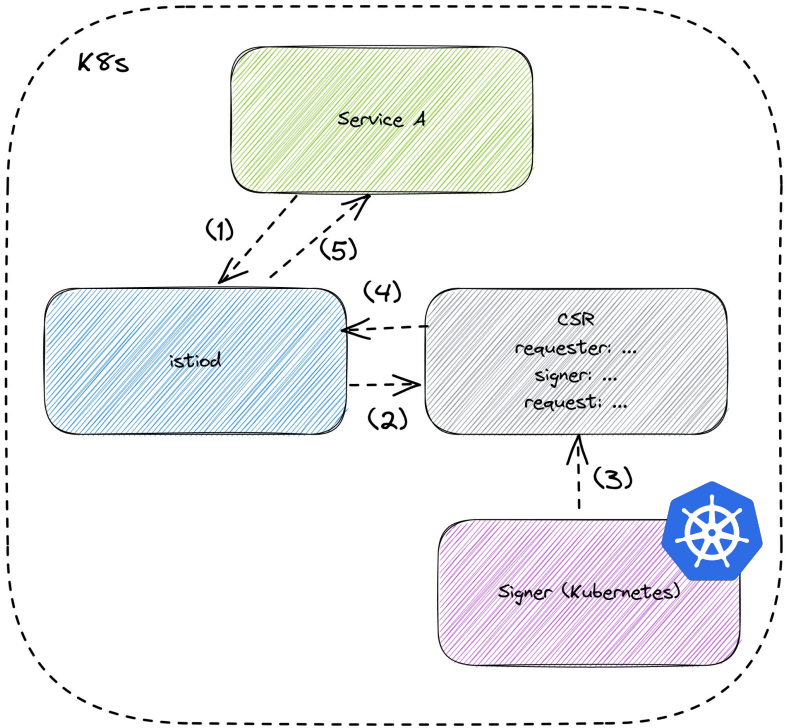
```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: isolated-issuer-root-csr
  annotations:
    experimental.cert-manager.io/request-duration: 1h0m0s
    experimental.cert-manager.io/request-is-ca: "false"
spec:
  groups:
  - system:masters
  - system:authenticated
  username: kubernetes-admin
  request: ...
  signerName: issuers.isolated-issuer.jetstack.io/istio-system.root-ca-issuer
  usages:
  - signing
  - key encipherment
status:
  certificate: ...
  conditions:
  - lastTransitionTime: "2021-08-02T18:23:52Z"
    lastUpdateTime: "2021-08-02T18:23:52Z"
    message: This CSR was approved by kubectl certificate approve.
    reason: KubectlApprove
    status: "True"
    type: Approved
```

Kubernetes CSR



1. CSR requested by Service A to istiod using gRPC
2. istiod creates CSR (requestor), Approves the request
3. Kubernetes signs request
4. istiod reads signed certificate
5. istiod responds to service with signed certificate

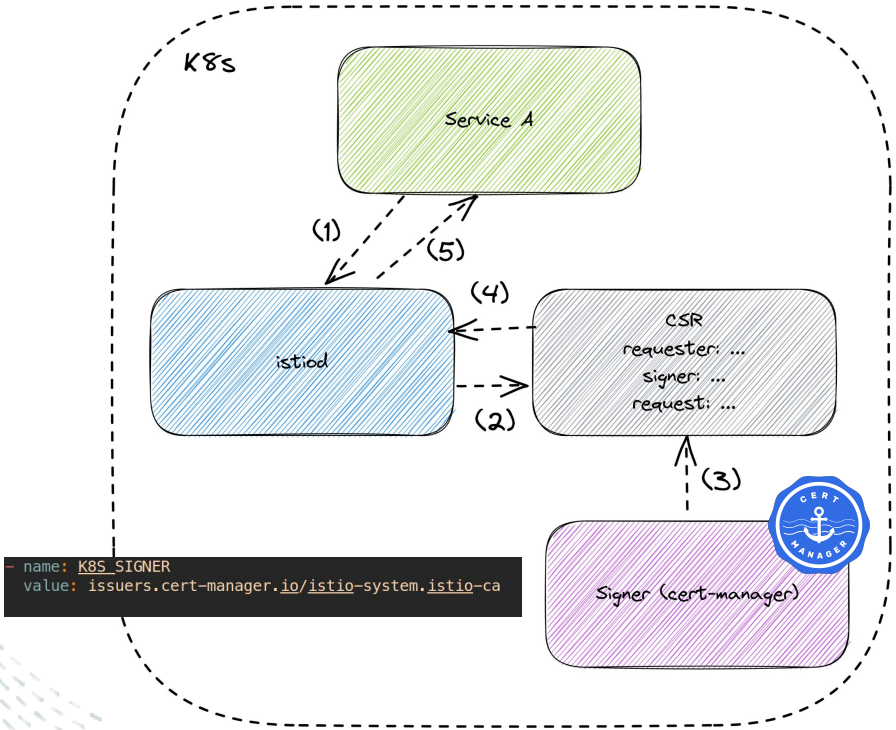
Kubernetes CSR



Using the Kubernetes control plane Trust Domain is problematic.

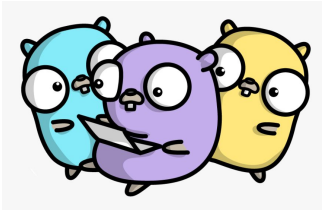
```
# Tells Istiod to use the Kubernetes legacy CA Signer
- name: K8S_SIGNER
  value: kubernetes.io/legacy-unknown
```

Kubernetes CSR with cert-manager



1. CSR requested by Service A to istiod using gRPC
2. istiod creates CSR (requestor), Approves the request
3. cert-manager signs request
4. istiod reads signed certificate
5. istiod responds to service with signed certificate

istio-csr



istio-csr acts as RA

cert-manager acts as CA

istiod doesn't serve as RA or CA

Many issuers supported

aws

KMS or PCA

Venafi

Google Cloud

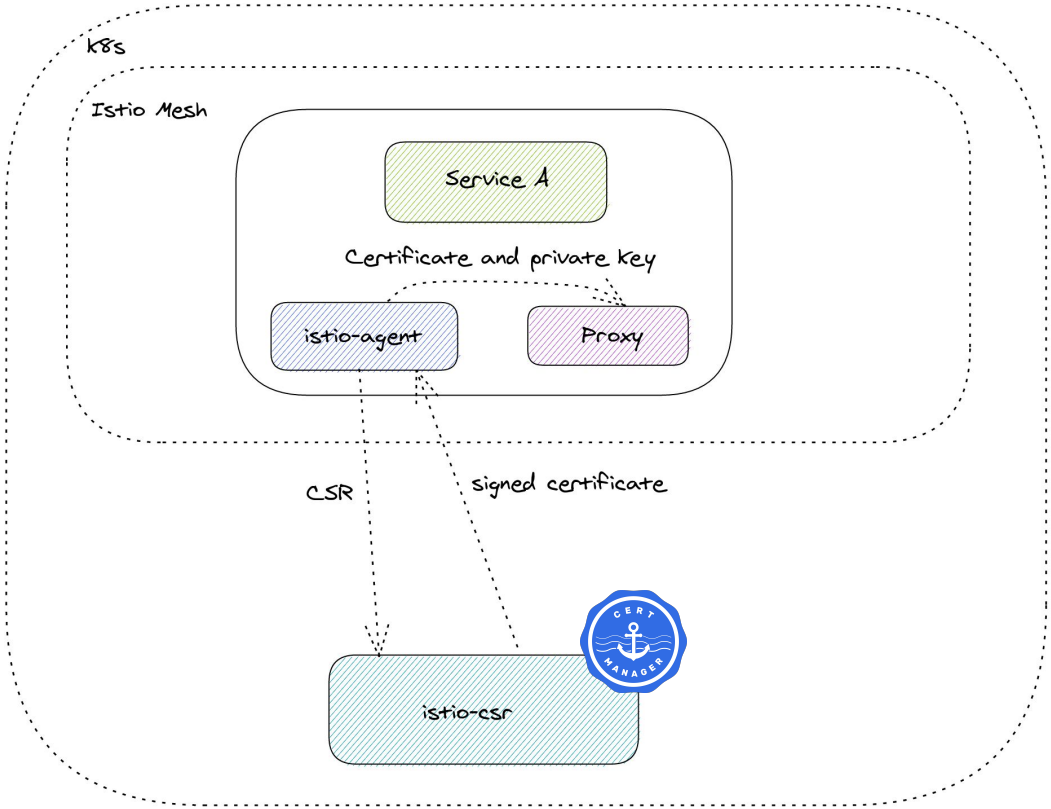
freeIPA
identity | policy | audit

smallstep

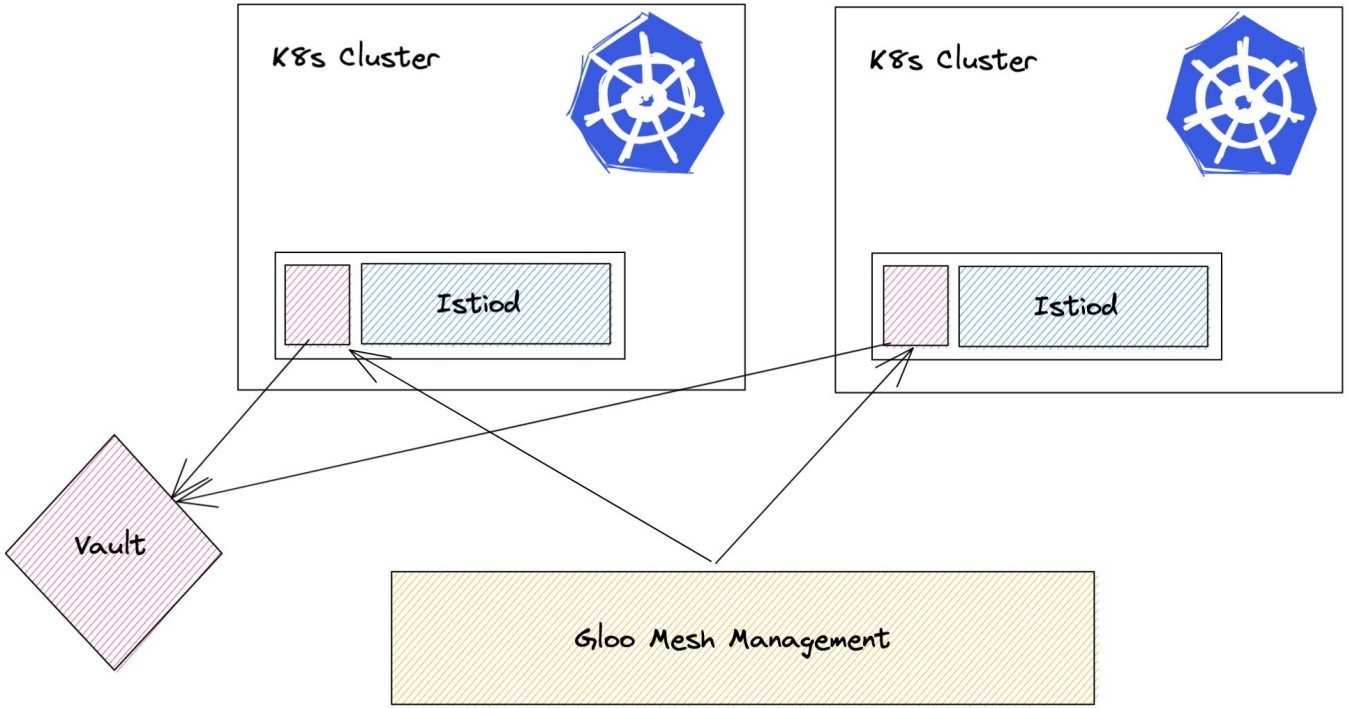
CFSSL

HashiCorp
Vault

istio-csr diagram



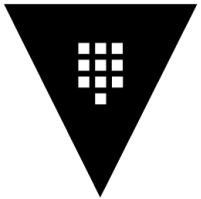
Another approach



DEMO

Vault configured as CA

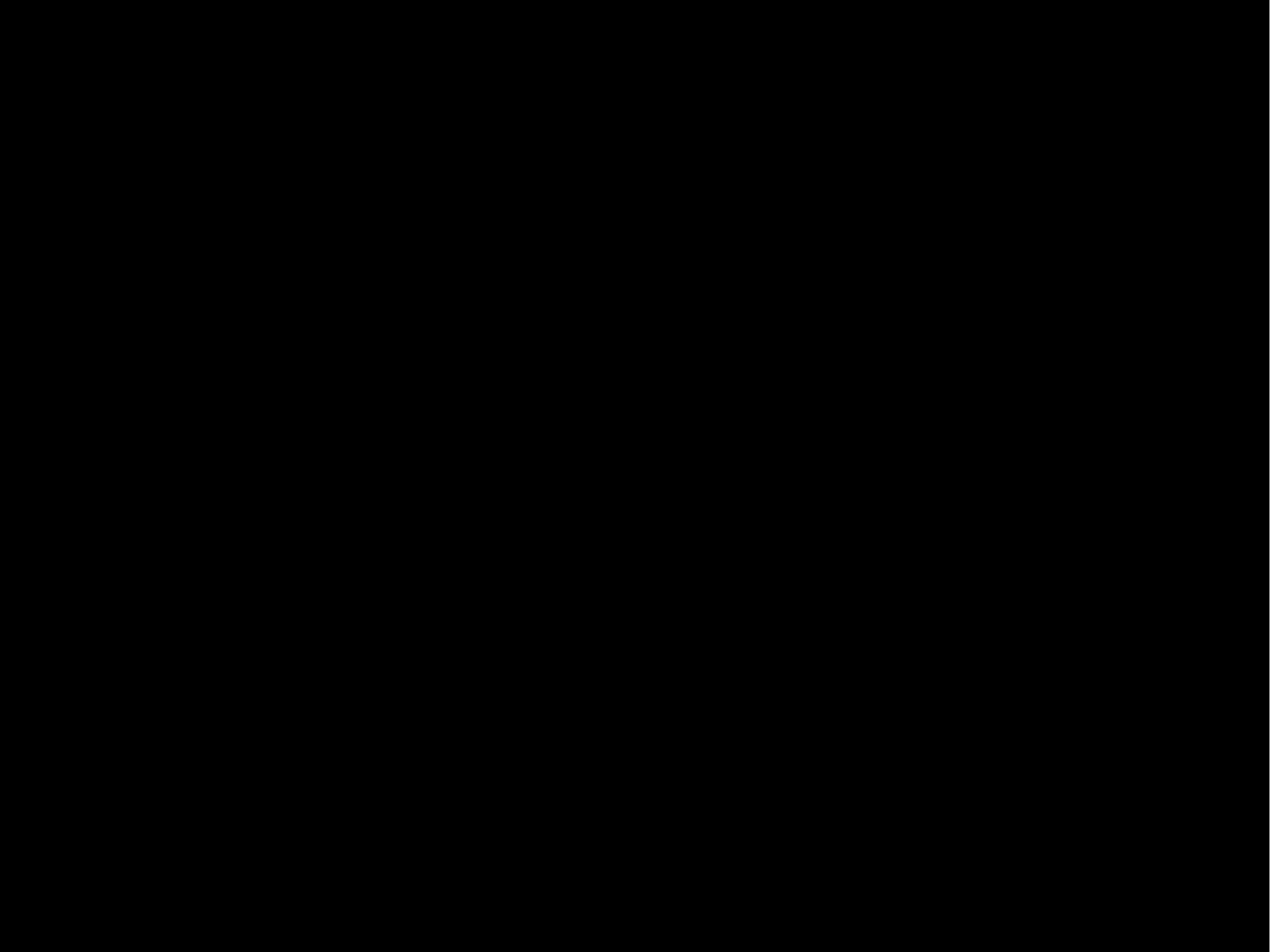
Install cert-manager, Vault issuer, and istio-csr



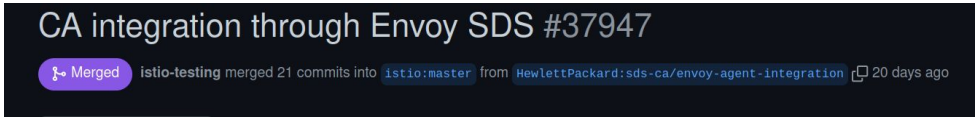
HashiCorp Vault

Install istio

Install bookinfo

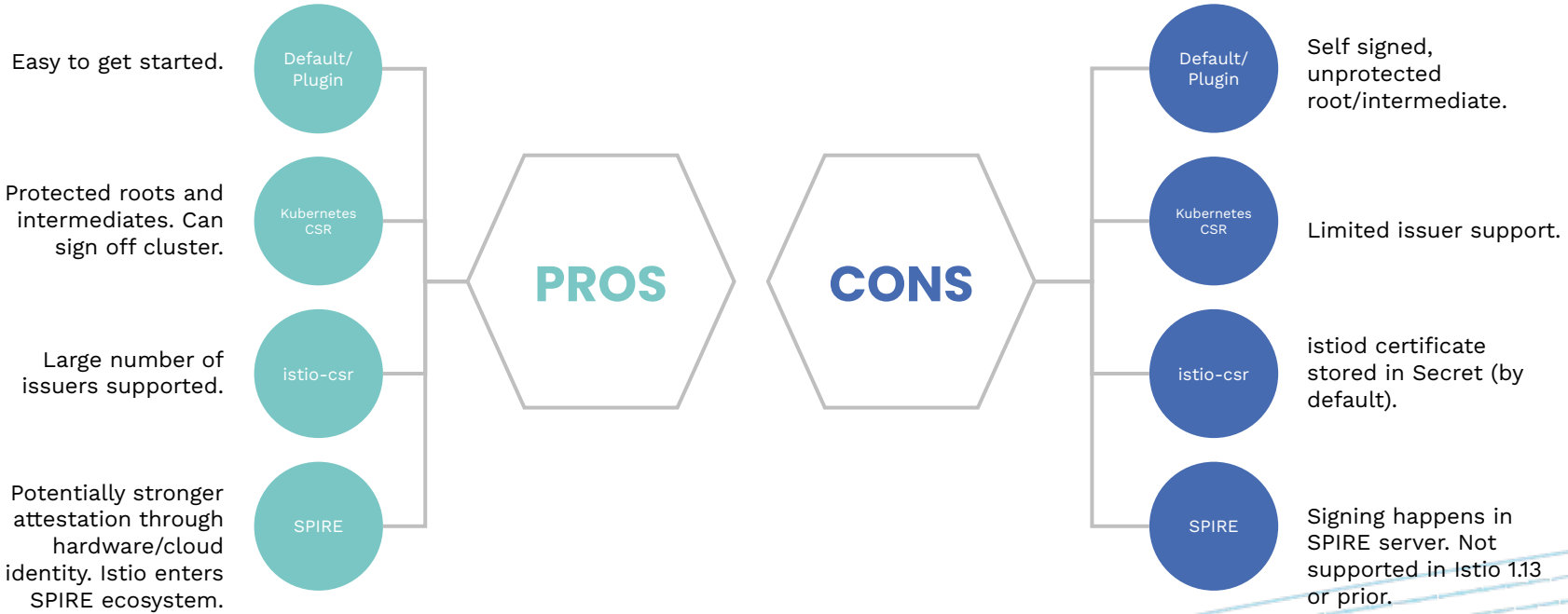


SPIRE



- The SPIFFE Runtime Environment
- Attests workloads, and issues SPIFFE identities (SVIDs)
- SPIRE server acts as CA (who's CA certificate can be minted from an "upstream authority")
- Pinned for istio v1.14 release

Pros & Cons


Easy to get started.
Default/ Plugin
Protected roots and intermediates. Can sign off cluster.
Kubernetes CSR
Large number of issuers supported.
istio-csr
Potentially stronger attestation through hardware/cloud identity. Istio enters SPIRE ecosystem.
SPIRE
PROS
CONS
Default/ Plugin
Self signed, unprotected root/intermediate.
Kubernetes CSR
Limited issuer support.
istio-csr
istiod certificate stored in Secret (by default).
SPIRE
Signing happens in SPIRE server. Not supported in Istio 1.13 or prior.

Thank you!

Director of Open Source, Solo.io

@linsun_unc

lin.sun@solo.io

linkedin.com/pub/lin-sun/1/…

Staff Software Engineer, Jetstack

@JoshVanL

joshua.vanleeuwen@jetstack.io

linkedin.com/in/joshvanl/