KubeCon | CloudNativeCon

North America 2023

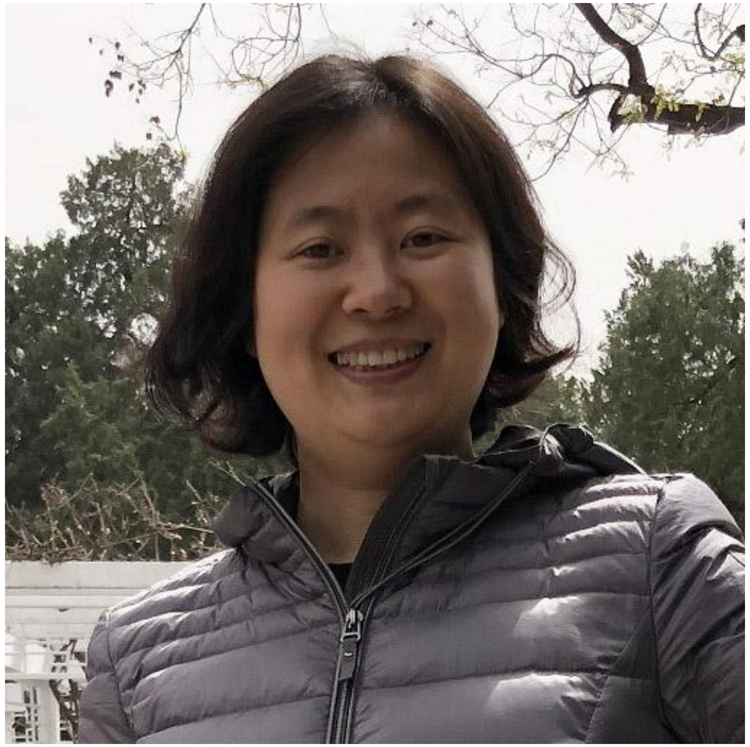
KubeCon | CloudNativeCon
North America 2023

Flavors of certificates in Service Mesh: The Whys and Hows!

Iris Ding - Intel

Faseela K - Ericsson Software Technology

Iris Ding



Cloud Software Engineer

intel

Istio

Steering Committee Member

Faseela K

Cloud Native Developer

Ericsson Software Technology

AMBASSADOR
CLOUD NATIVE
COMPUTING FOUNDATION

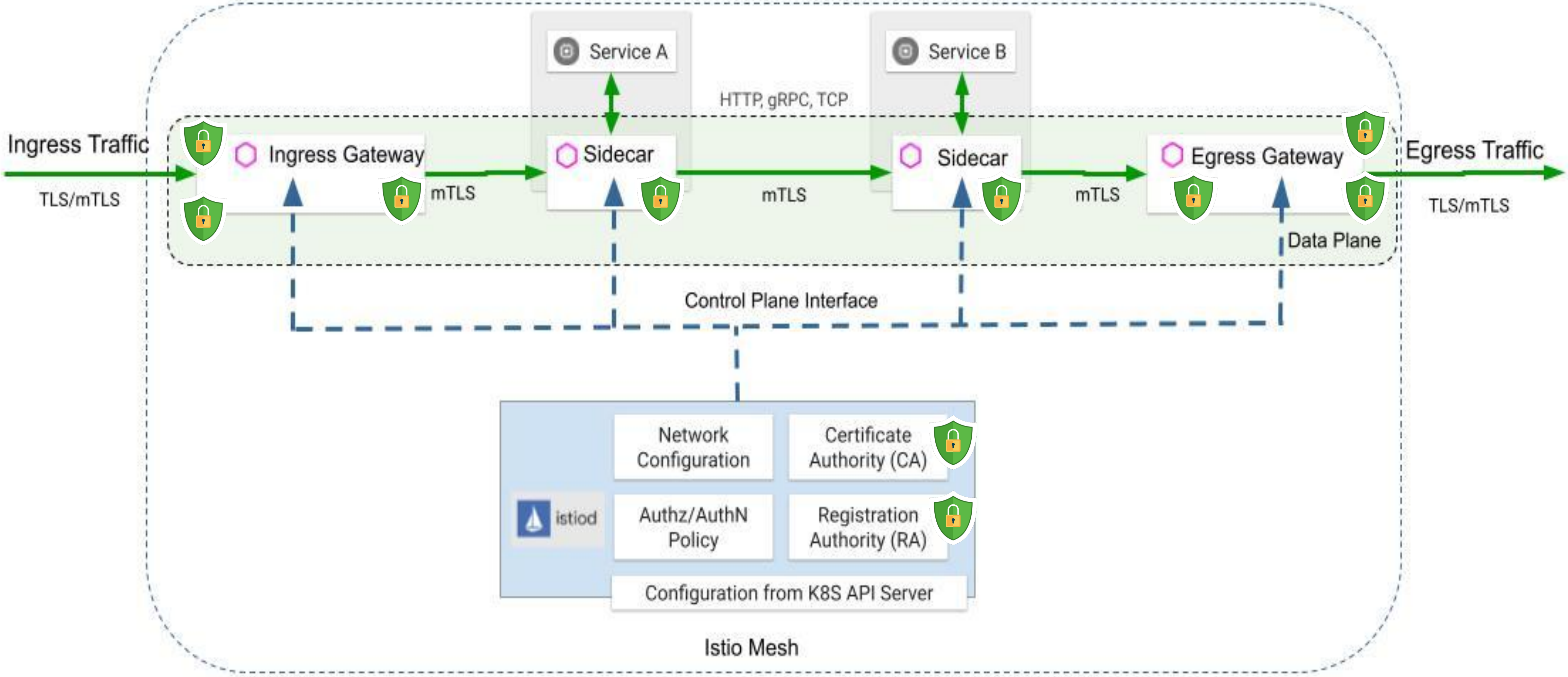
Istio

Steering Committee Member

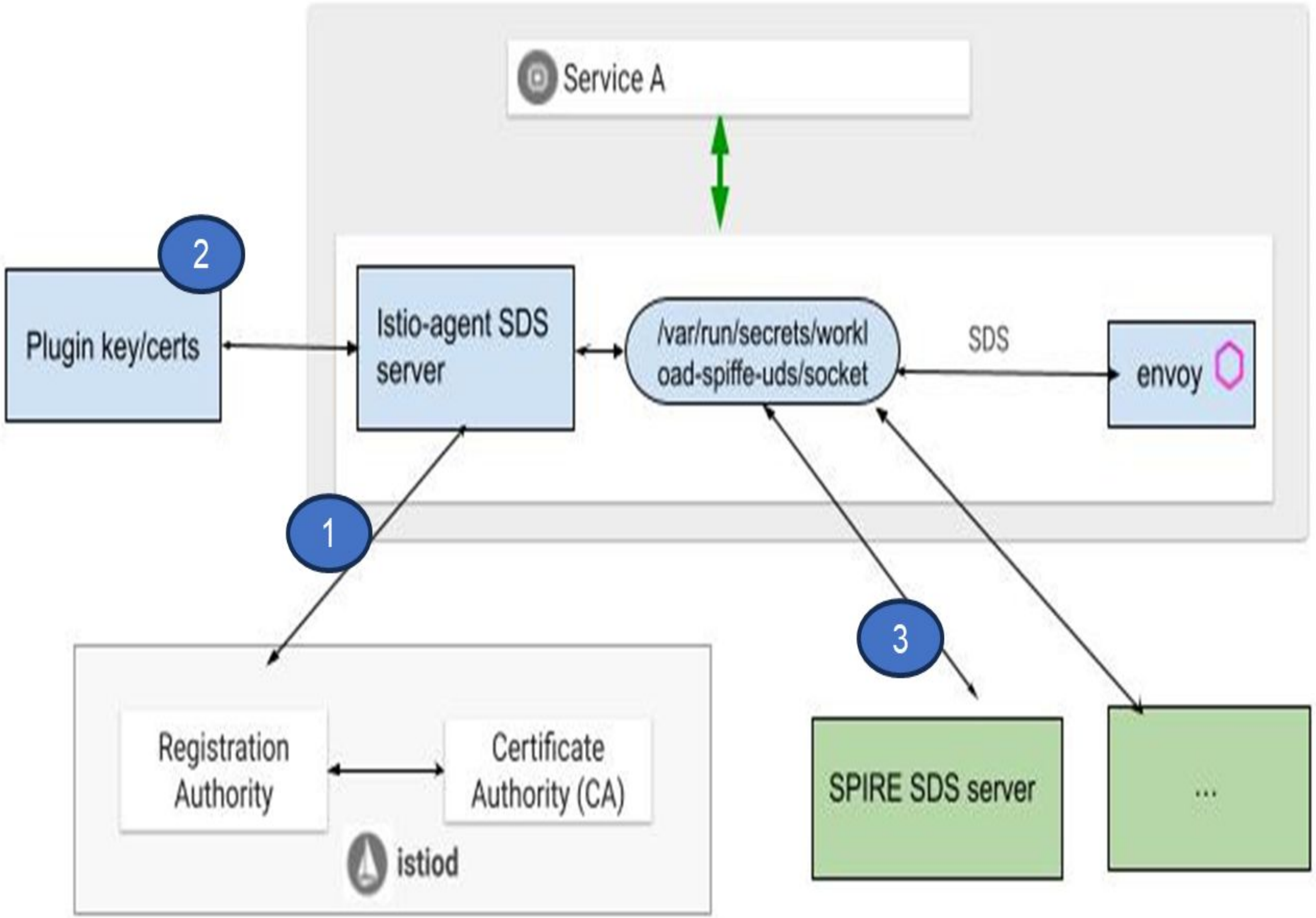
Agenda

- Certificates in Istio Service mesh overview
- Workload certificate
- Certificate Authority certificate
- Confidential computing
- Certificate Revocation List, OCSP stapling
- extended TLS settings
- 5G Telco Security Overview

Certificates in Istio Service Mesh



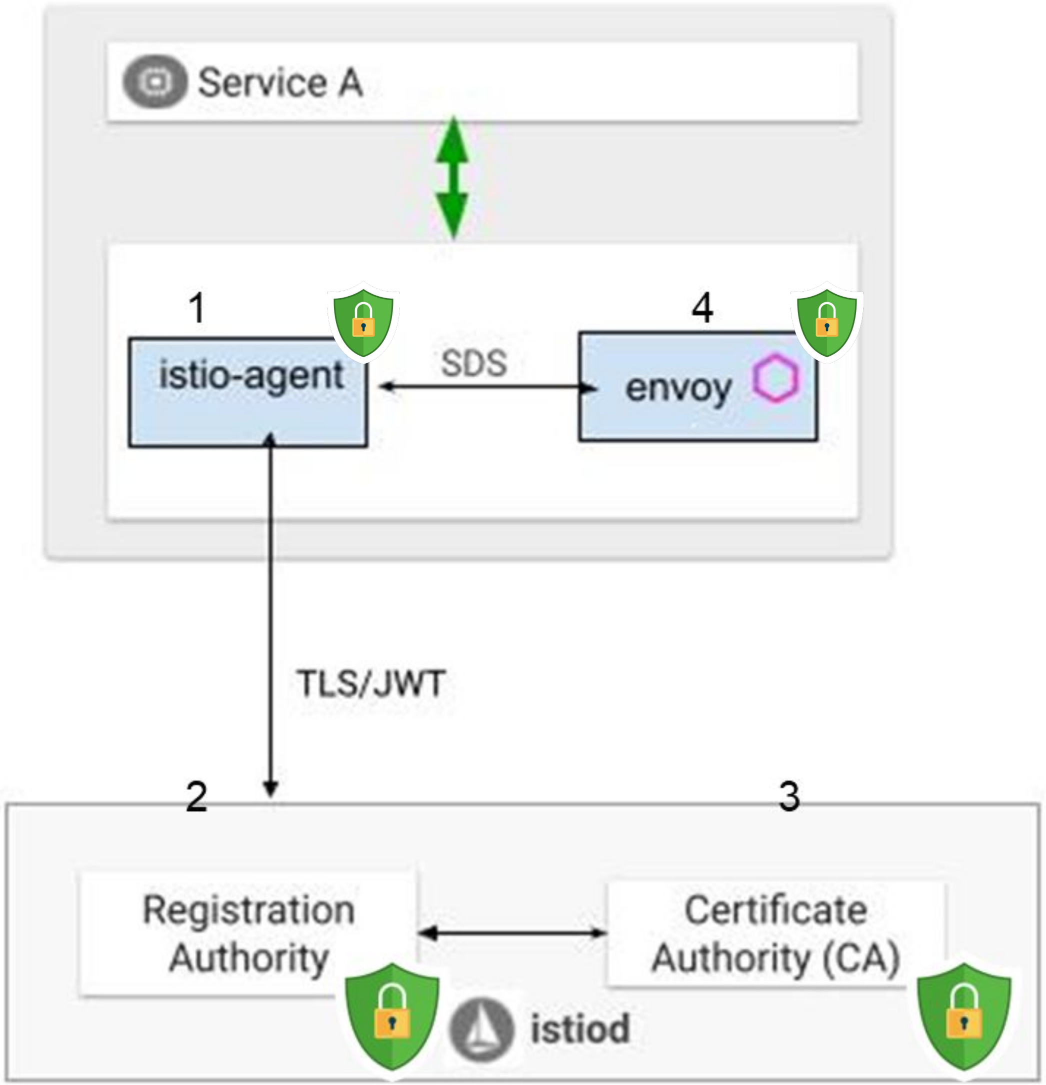
Workload Certificate



Workload certificate options:

1. Istio-agent as SDS Server & CA client
2. Plug-in key & certificate
3. External SDS server

Workload Certificate



Steps:

1. Istio-agent generate private key/CSR and send CSR to Registration Authority(RA) in Istiod

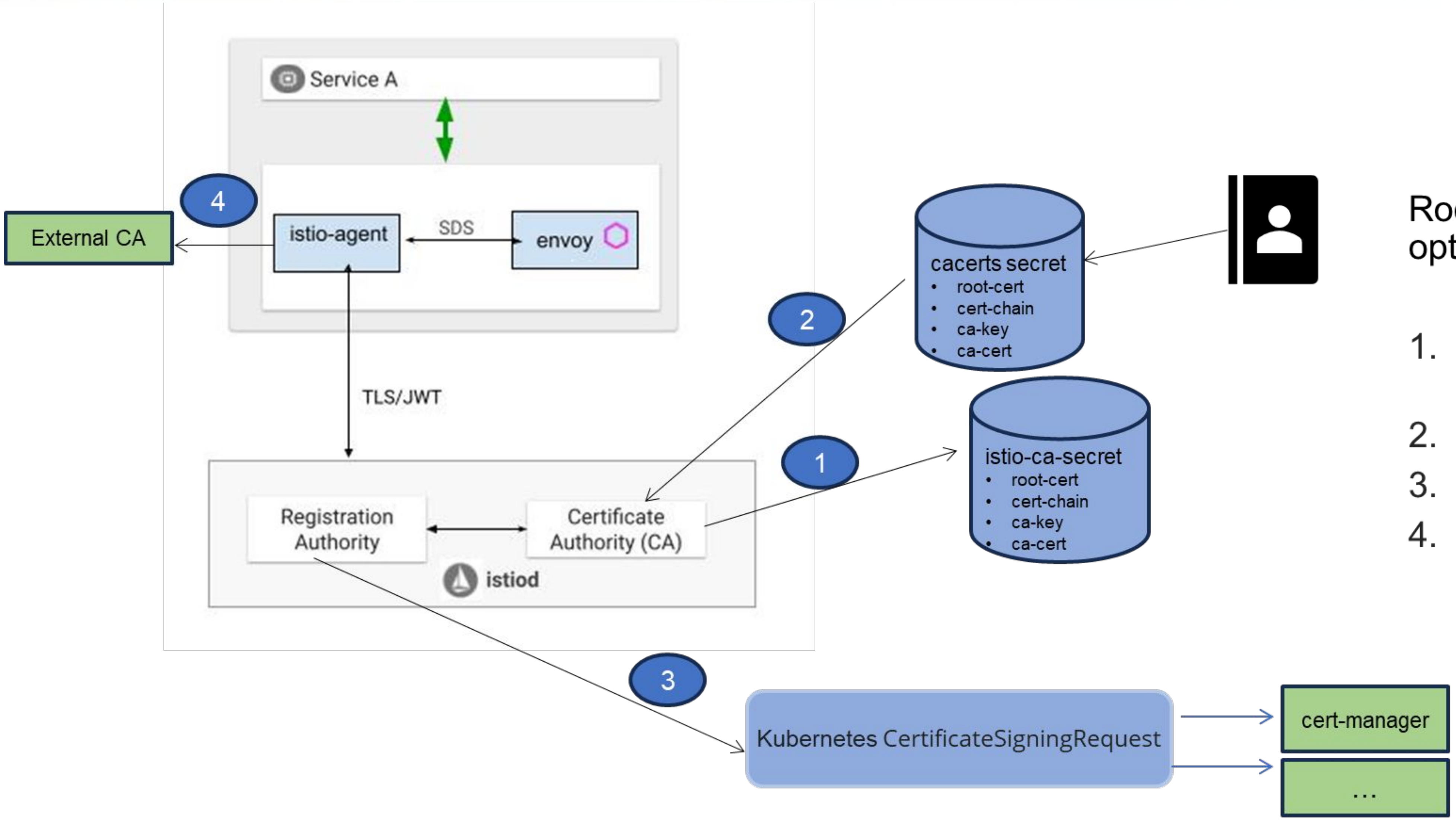
2. RA Authenticate CSR

3. Certificate Authority Sign CSR

4. Envoy get private key and cert

RA's certificate and CA's certificate can be different

Certificate Authority Server Certificate



Service A

4

External CA

istio-agent

SDS

envoy

TLS/JWT

Registration Authority

Certificate Authority (CA)

istiod

2

cacerts secret
- root-cert
- cert-chain
- ca-key
- ca-cert

1

istio-ca-secret
- root-cert
- cert-chain
- ca-key
- ca-cert

3

Kubernetes CertificateSigningRequest

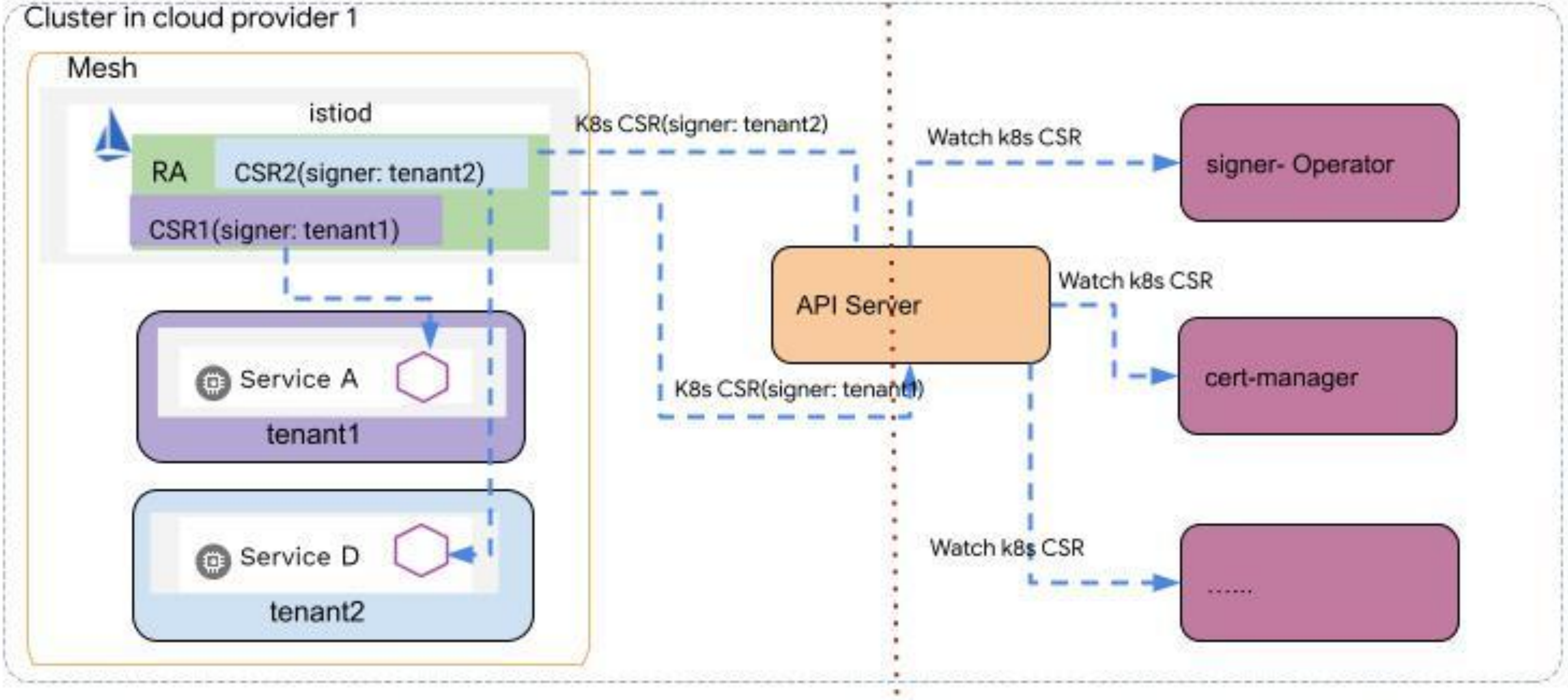
cert-manager

…

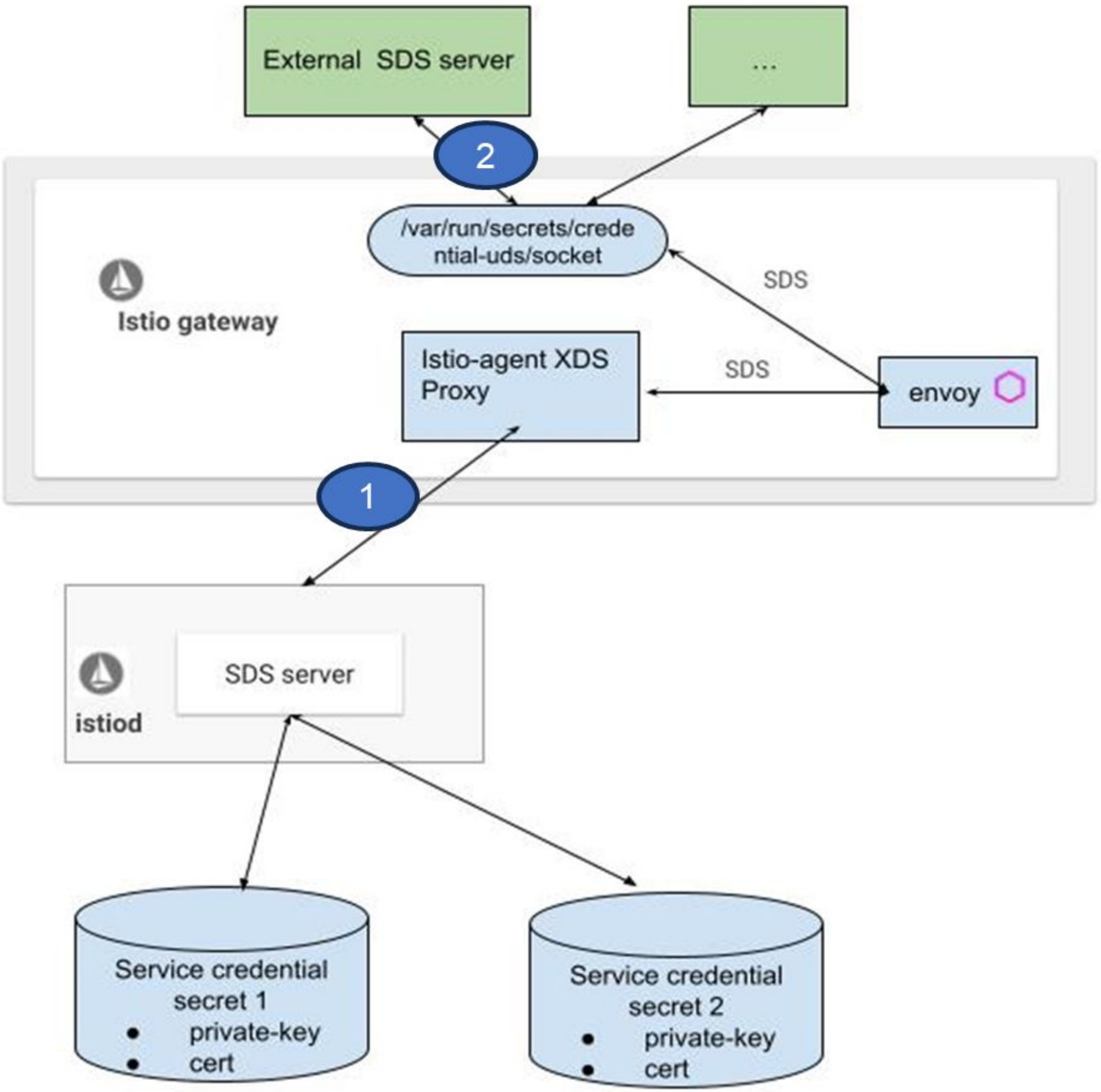
Root of trust for the mesh options:

1. Self-Signed CA certificate
2. Plugin CA certificate
3. Kubernetes CSR
4. External CA

Multiple CA


Cluster in cloud provider 1
Mesh
istiod
RA
CSR2(signer: tenant2)
CSR1(signer: tenant1)
Service A
tenant1
Service D
tenant2
K8s CSR(signer: tenant2)
K8s CSR(signer: tenant1)
API Server
Watch k8s CSR
signer- Operator
Watch k8s CSR
cert-manager
Watch k8s CSR
......

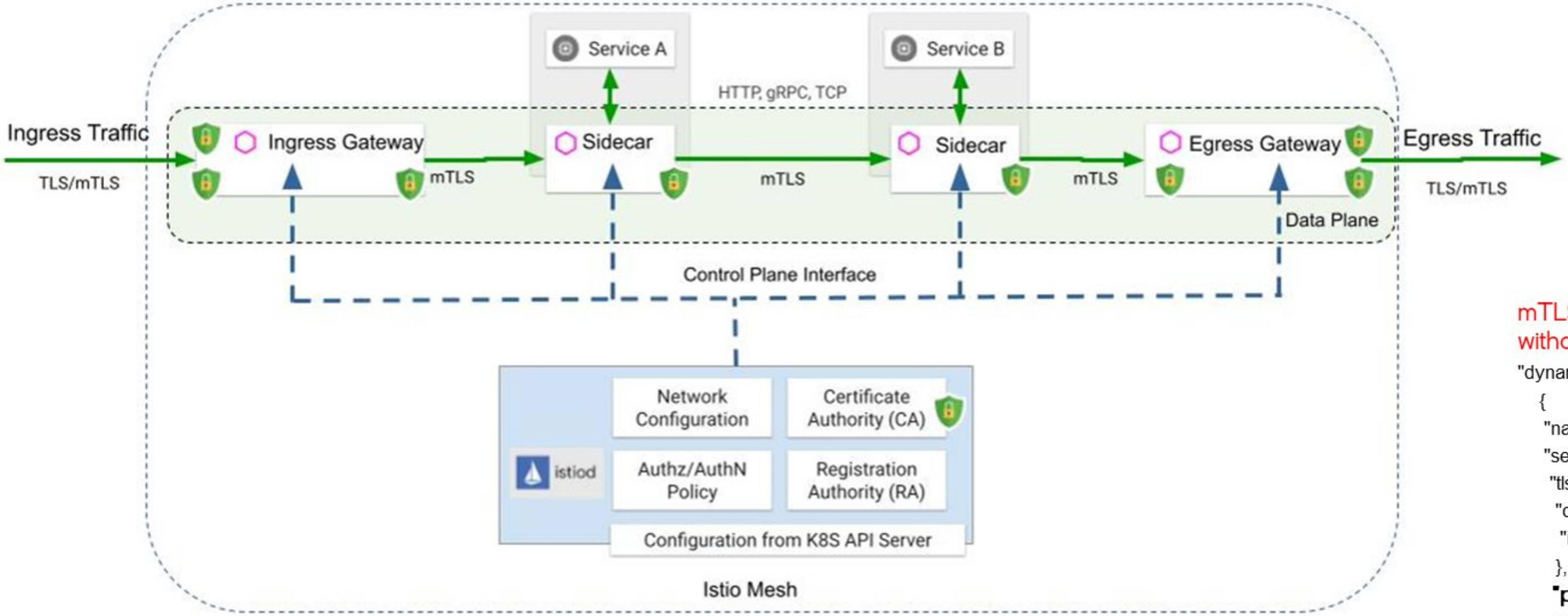
Gateway Certificates



Gateway Certificates options:

1. Istiod as SDS Server
2. External SDS server

Secured Enough?



mTLS private key is saved in sidecar memory without encryption

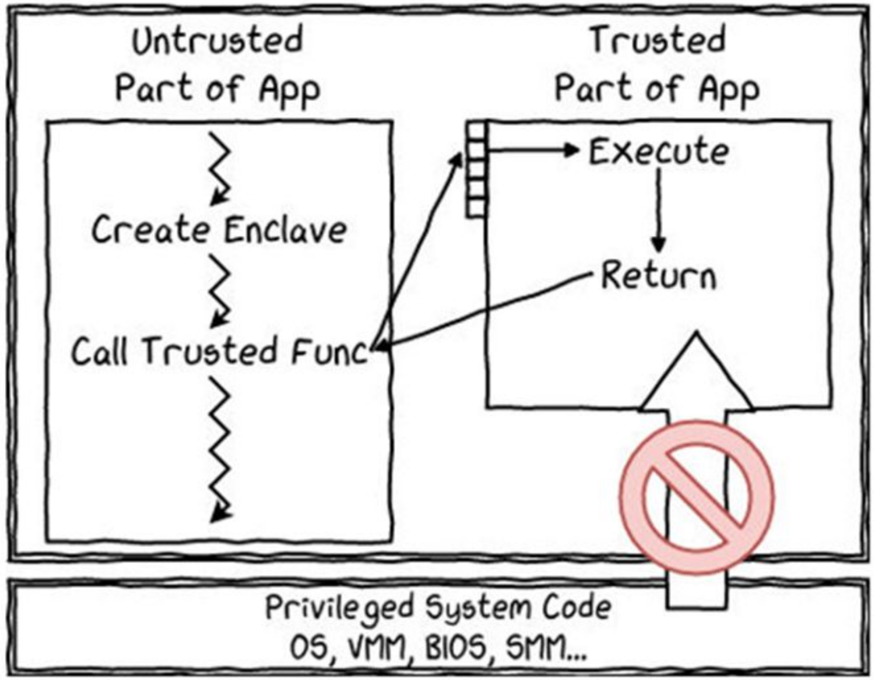
```
"dynamic_active_secrets":  [
   {
    "name":  "default",
    "secret": {
     "tls_certificate": {
      "certificate_chain": {
       "inline_bytes":  "LS0tLS1CR...=="
      },
     "private_key":{
      "inline_bytes":"W3JIZGFjdGVkXQ==" //[redacted]
 ...
```

Private keys for istiod CA server and gateway are in clear text in k8s secret

```
kubectl get secret istio-ca-secret -n istio-system -o jsonpath='{.data.ca-key\.pem}' |base64 –d
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAwzwqi2HvLm5XY1eHzKbx8Q5KxgJfGP/zMQb0PAyaN1/XHh1f
s7xh5MHPx+tSJ9tJW2ShwyBN9VoRtb/QMYjGSNv0e18v+9+bpeduhr5CtlgSWTo2
oV+UMtRZYgXfuNxNIY7eL4lJ9OUyDU5ELJYZHFDUkIR0jJ47LKNxzDiEXwn0sZ++
zxwhL3glp2ccQ55eC2azj1WRurSsjf7Vq1TeSlhOyiFUzSAGSN9DHTk6U5AMX9Fr
Nh2zuJ5JtnD2wxyttogTipXL0PLtixXt7zwt6bdUZXOgilFCZlEU5ODuLDtpdMCd
```

Key protection via Intel® SGX

Intel® SGX – a process-based TEE

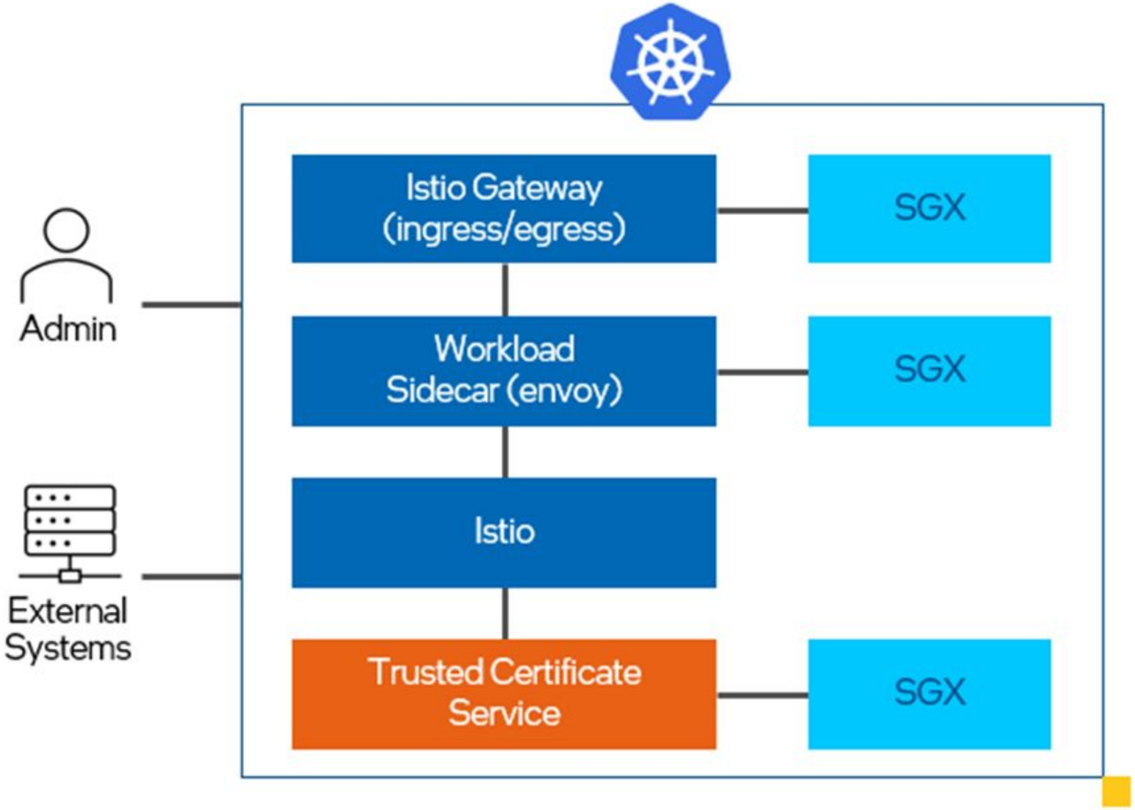


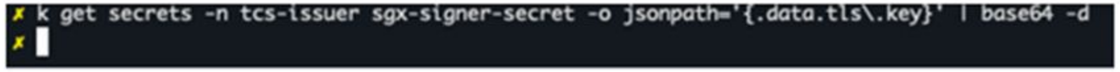
Dependency

Intel® Software Guard Extensions

Platform

3rd Gen Intel® Xeon® processor
4th Gen Intel® Xeon® processor



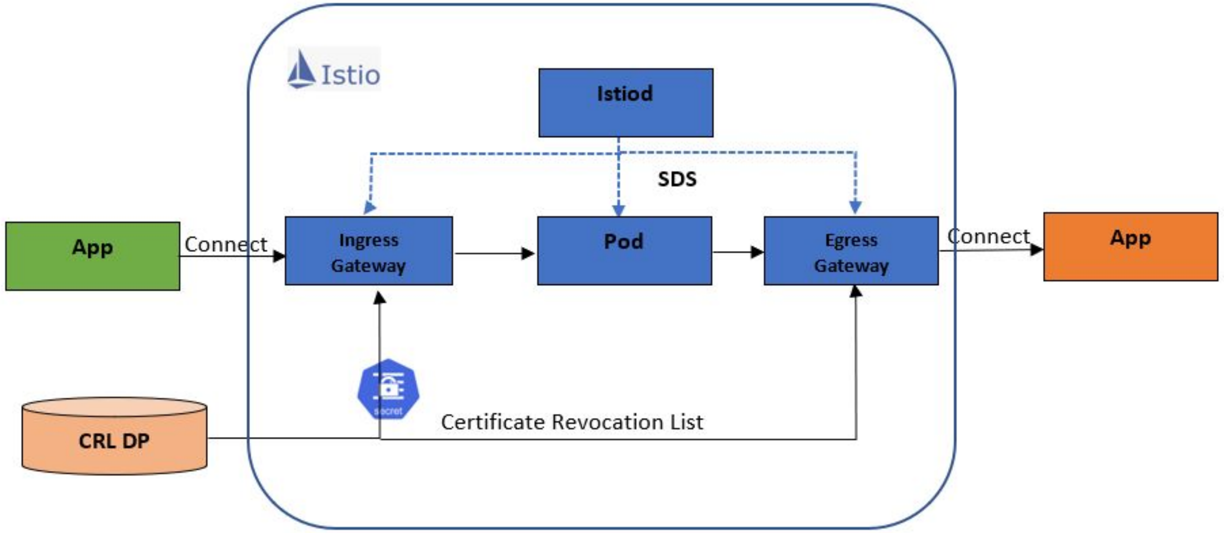


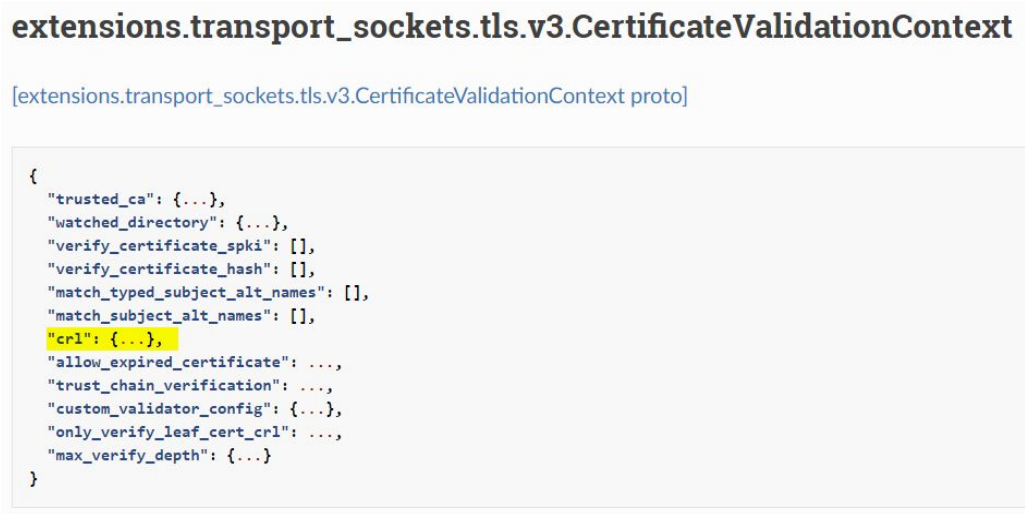
```
✗ k get secrets -n tcs-issuer sgx-signer-secret -o jsonpath='{.data.tls\.key}' | base64 -d
✗ █
```

Reference:

- hsm-sds-server
- trusted-certificate-issuer

Certificate Revocation Lists(CRL)





extensions.transport_sockets.tls.v3.CertificateValidationContext

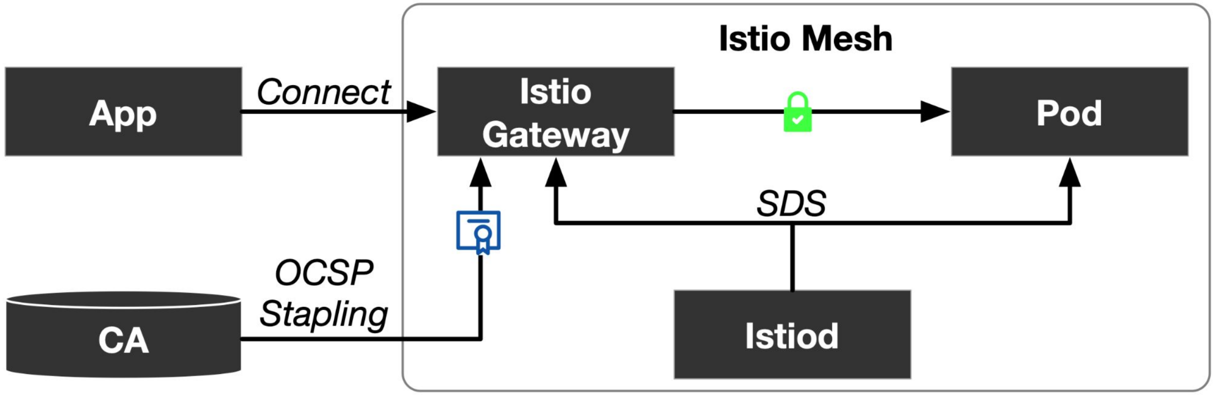
[extensions.transport_sockets.tls.v3.CertificateValidationContext proto]

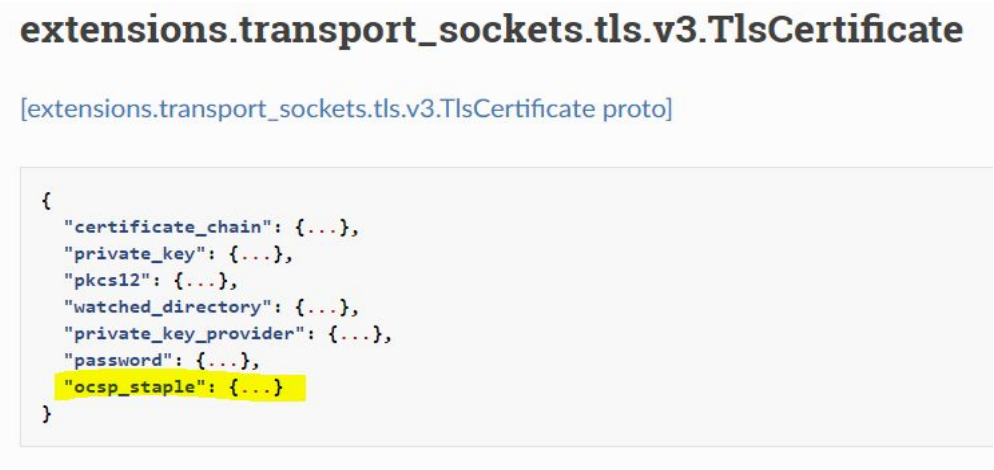
```
{
  "trusted_ca": {...},
  "watched_directory": {...},
  "verify_certificate_spki": [],
  "verify_certificate_hash": [],
  "match_typed_subject_alt_names": [],
  "match_subject_alt_names": [],
  "crl": {...},
  "allow_expired_certificate": ...,
  "trust_chain_verification": ...,
  "custom_validator_config": {...},
  "only_verify_leaf_cert_crl": ...,
  "max_verify_depth": {...}
}
```

- CRLs are lists of revoked certificates maintained by the CAs, accessible through a CRL Distribution Point (CDP).
- As per the 3GPP 5G Telecom Standard, the specification states that CRL status check should be supported for external certificate validation
- The user may provide CRL datasource in the same Secret as the TLS Certificate and Key, or as a separate secret.
- K8s secrets are limited in size, so there will be an inherent limitation to the CRL bundle in this design.

```
kind: Secret
metadata:
  name: client-crt
data:
  tls.crt:RndvR1pYSXZZWGR6RUZvYURPStlc2wxSOMEBYTES
  tls.key: enlhd3FlVGJhd0xZOTHERBYTES
  ca.crl: MjAyMy0wMi0wMSAxNjo1NOCSPBYTES
```

OCSP Stapling





extensions.transport_sockets.tls.v3.TlsCertificate

[extensions.transport_sockets.tls.v3.TlsCertificate proto]

```
{
  "certificate_chain": {...},
  "private_key": {...},
  "pkcs12": {...},
  "watched_directory": {...},
  "private_key_provider": {...},
  "password": {...},
  "ocsp_staple": {...}
}
```

- OCSP is a protocol used to check the revocation status of individual certificates
- The OCSP response to be stapled with this certificate during the handshake.
- The response must be DER-encoded and may only be provided via filename or inline_bytes
- The user is expected to provide a pre-fetched OCSP staple in the same Secret as the TLS Certificate and Key.

```
kind: Secret
metadata:
  name: server-crt
data:
  tls.crt:RndvR1pYSXZZWGR6RUZvYURPStlc2wxSOMEBYTES
  tls.key: enlhd3FlVGJhd0xZOTHERBYTES
  tls.ocsp: MjAyMy0wMi0wMSAxNjo1NOCSPBYTES
```

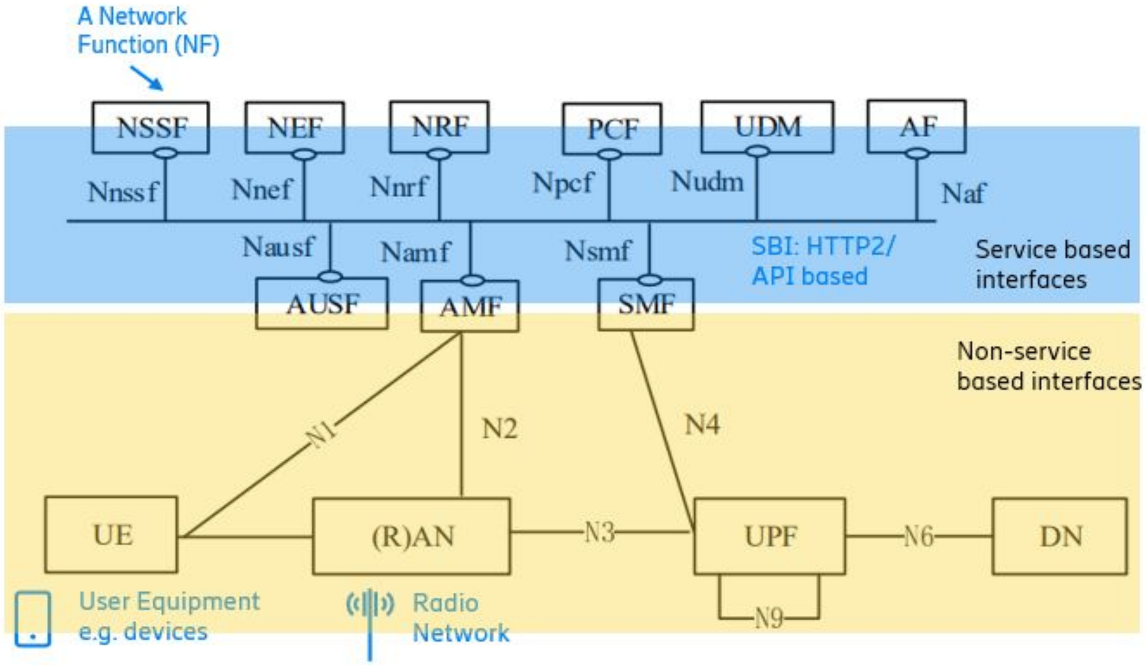
Extended TLS Settings

- Cipher Suites
  - Mesh-wide/Mesh-external (ignored for TLS1.3)
- ECDH Curves
  - Mesh-external configuration
- Signature Schemes
  - Currently supported only via EnvoyFilter

Reference: Support For Extended TLS Settings · Issue #41645 · istio/istio (github.com)
Common TLS configuration (proto) — envoy 1.29.0-dev-75d395 documentation (envoyproxy.io)

5G System Architecture

- Separate Control Plane and User Plane
- Network Functions (NFs) are defined by 3GPP standard and consist of smaller unit functions called NF services
- NFs implemented using cloud-native design principles are called Cloud-native NFs (CNFs)
- Different NFs connect to each other via uniform interface, called service-based interface (SBI), HTTP2/API based
- Service mesh is most useful for Control Plane NFs (HTTP2 based, less strict requirements on latency)

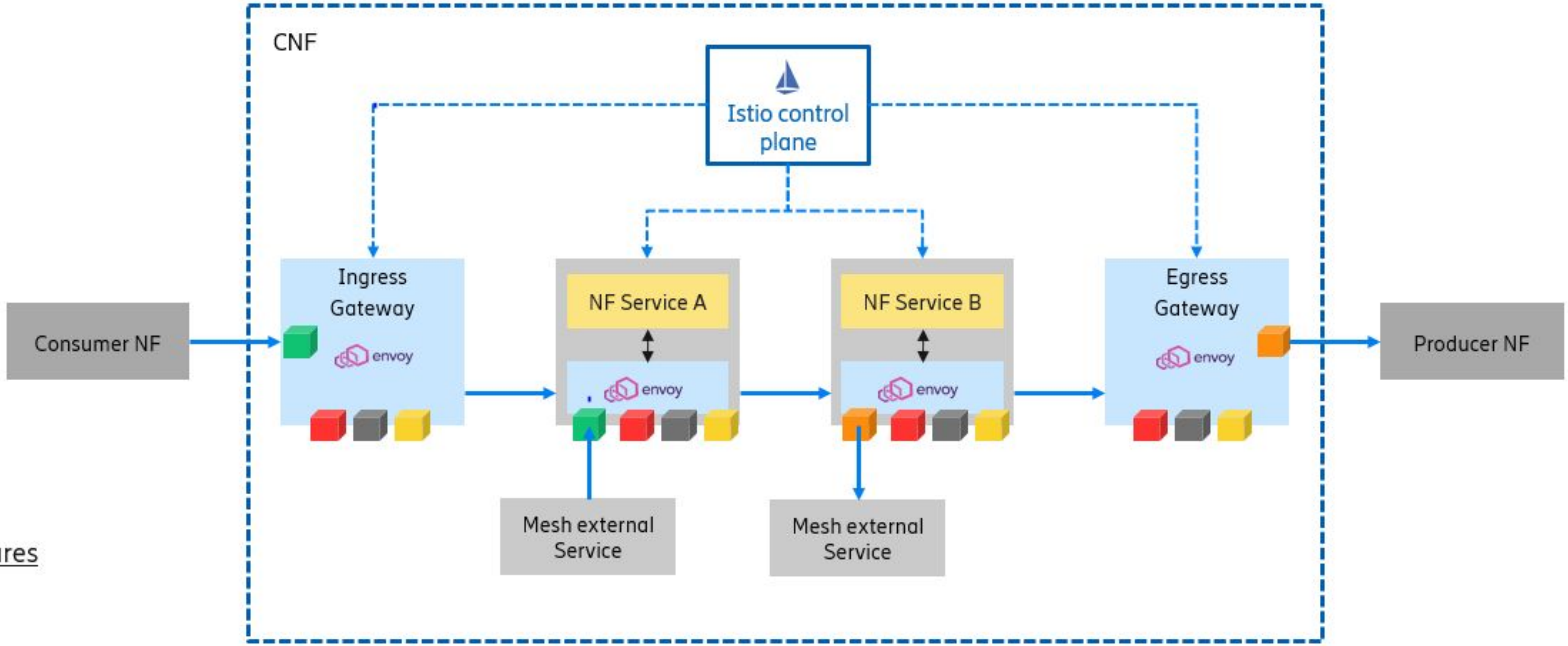


AUSF	Authentication Server Function
AMF	Access and Mobility Management Function
AF	Application Function
DN	Data Networks
SMF	Session Management Function
UDM	Unified Data Management

NEF	Network Exposure Function
NSSF	Network Slice Selection Function
NRF	Network Repository Function
PCF	Policy Control Function
UPF	User Plane Function
(R)AN	(Radio) Access Network

Reference: 3GPP TS 123 501 V15.3.0

Istio usage in a CNF



Service Mesh features

🟩 = Ingres

🟧 = Egress

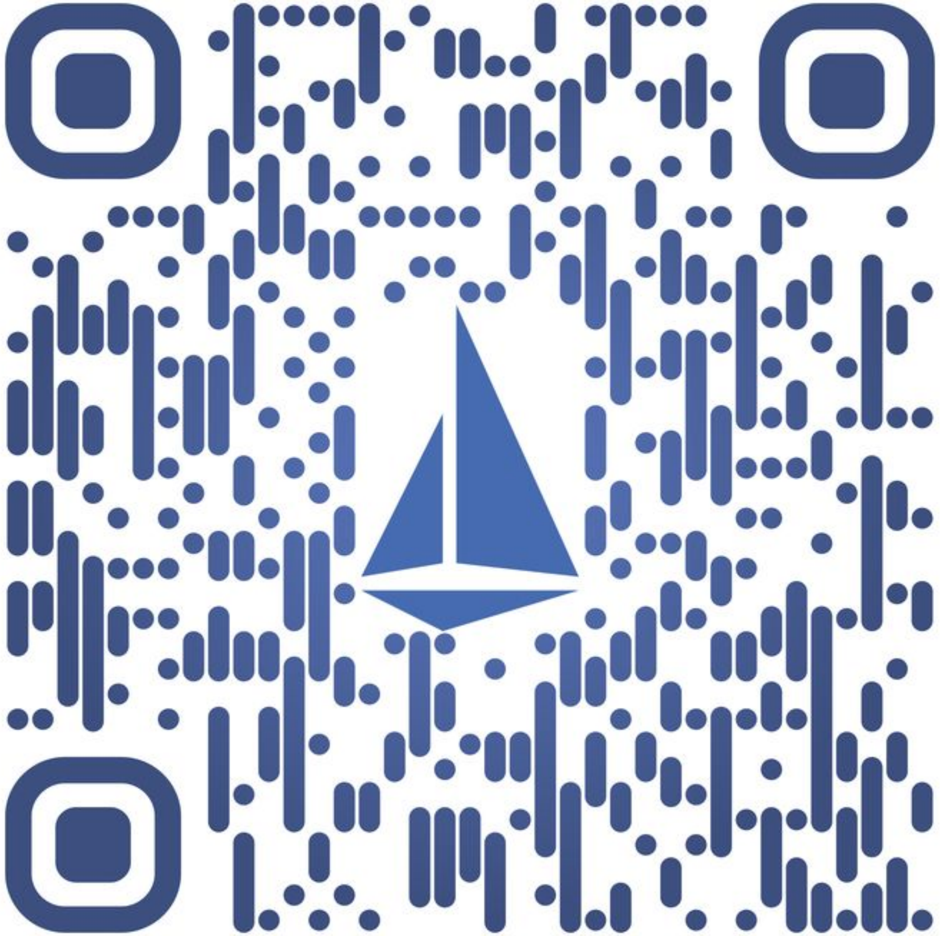
🟥 = Traffic Management

🟨 = Security

⬛ = Observability

Ingress/Egress Certificate Handling

- **External to ServiceMesh**
  - Certificates are handled at Ingress Gateway
  - Ingress TLS is configured at the Istio Gateway CR or alternatively k8s Gateway
- **ServiceMesh to External**
  - Certificates are handled at Egress Gateway or Egress Pods
  - Egress TLS is configured at the DestinationRule CR (alternative secrets can be mounted)
- **ServiceMesh to Cluster Internal**
  - Certificates are handled at Egress Pods
  - Egress TLS is configured at the DestinationRule CR
- **Cluster Internal to ServiceMesh**
  - Certificates are handled at the sidecar proxy
  - Ingress TLS is configured at the Istio Sidecar CR



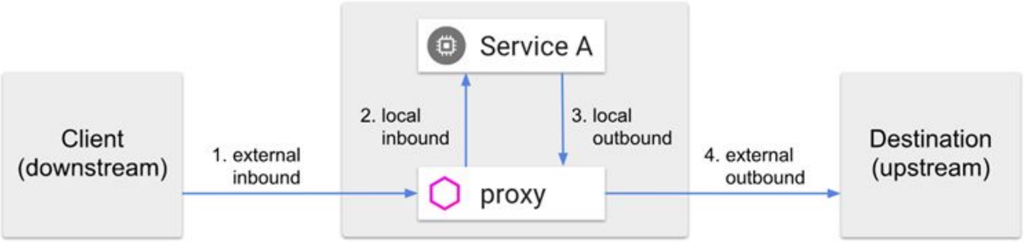
Please scan the QR Code above
to leave feedback on Istio

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

Thank You!

Technical Details

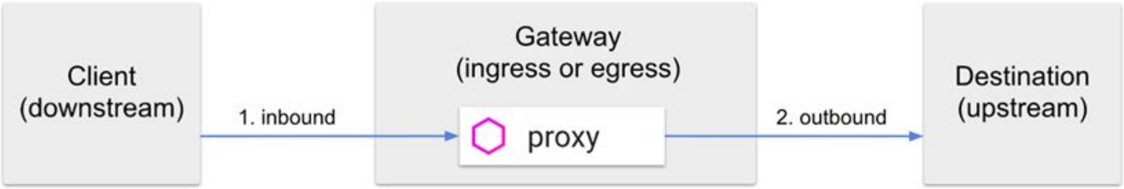
- Sidecars
  - PeerAuthentication is used to configure what type of mTLS traffic the sidecar will accept.
  - DestinationRule is used to configure what type of TLS traffic the sidecar will send.
  - Port names, or automatic protocol selection, determines which protocol the sidecar will parse traffic as.



Sidecar proxy network connections

- Gateways
  - The inbound request, initiated by some client such as curl or a web browser. This is often called the "downstream" connection.
  - The outbound request, initiated by the gateway to some backend. This is often called the "upstream" connection.

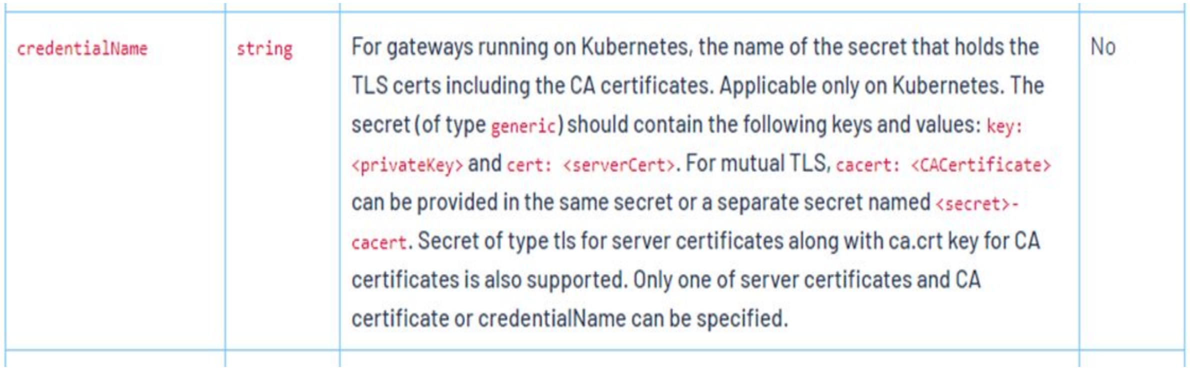
Both of these connections have independent TLS configurations.



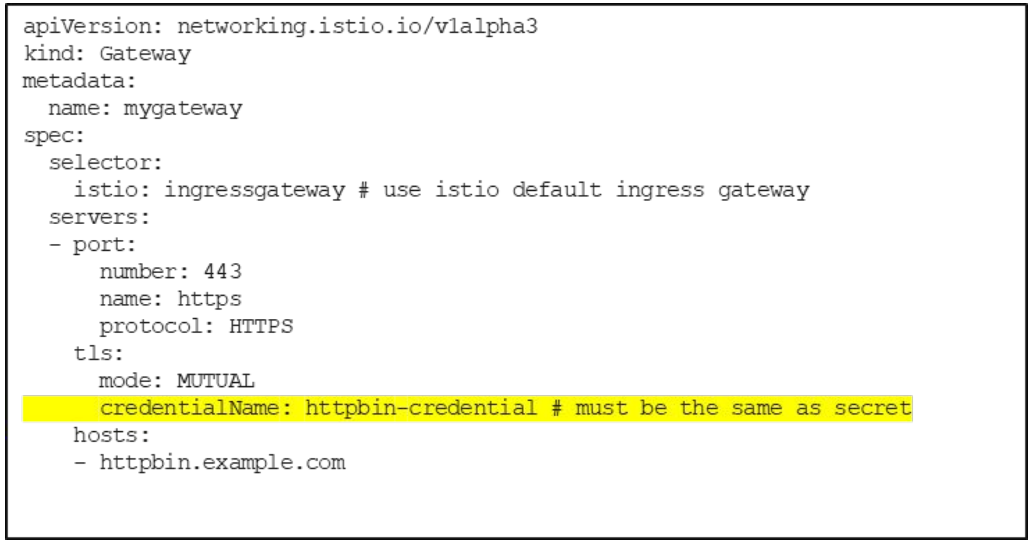
Gateway network connections

Technical Details

- Certificates are handled at Ingress Gateway
- Ingress TLS is configured at the Istio Gateway CR or alternatively k8s Gateway CR
- Secret name(s) must have format according to:



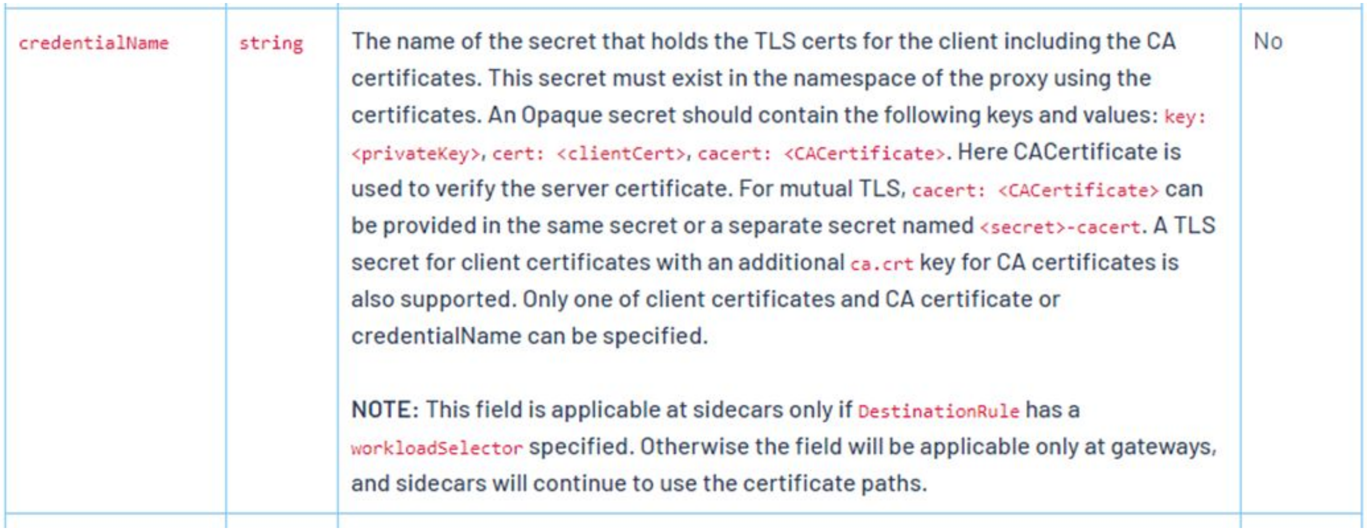
credentialName	string	For gateways running on Kubernetes, the name of the secret that holds the TLS certs including the CA certificates. Applicable only on Kubernetes. The secret (of type generic) should contain the following keys and values: key: <privateKey> and cert: <serverCert>. For mutual TLS, cacert: <CACertificate> can be provided in the same secret or a separate secret named <secret>-cacert. Secret of type tls for server certificates along with ca.crt key for CA certificates is also supported. Only one of server certificates and CA certificate or credentialName can be specified.	No



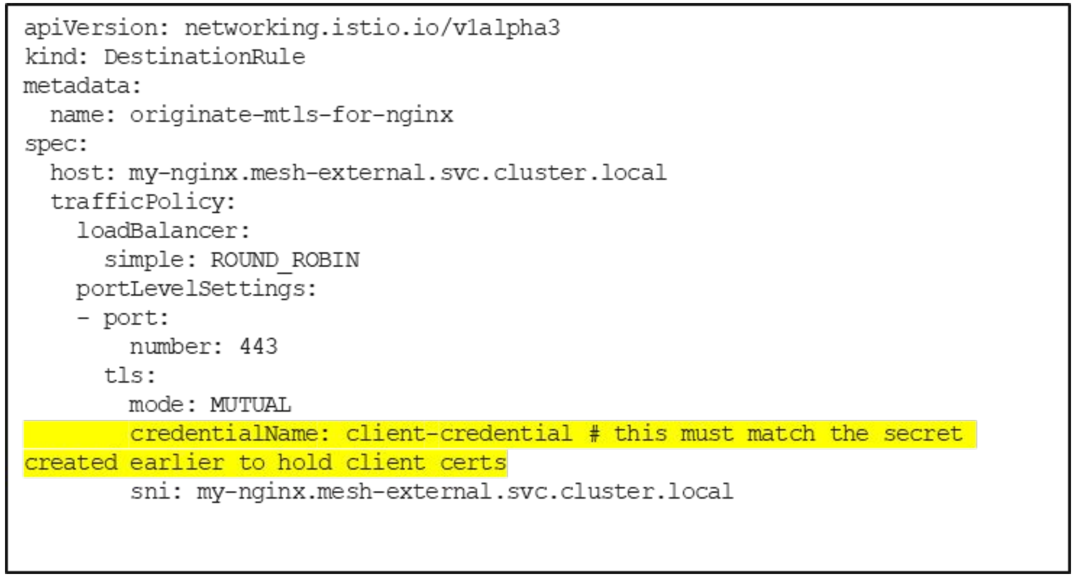
```yaml
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: mygateway
spec:
  selector:
    istio: ingressgateway # use istio default ingress gateway
  servers:
  - port:
      number: 443
      name: https
      protocol: HTTPS
    tls:
      mode: MUTUAL
      credentialName: httpbin-credential # must be the same as secret
    hosts:
    - httpbin.example.com
```

Technical Details

- Certificates are handled at Egress Gateway or Egress Pods
- Egress TLS is configured at the DestinationRule CR (alternative secrets can be mounted)
- Configuration example from Istio
- Credential name must have format according to:



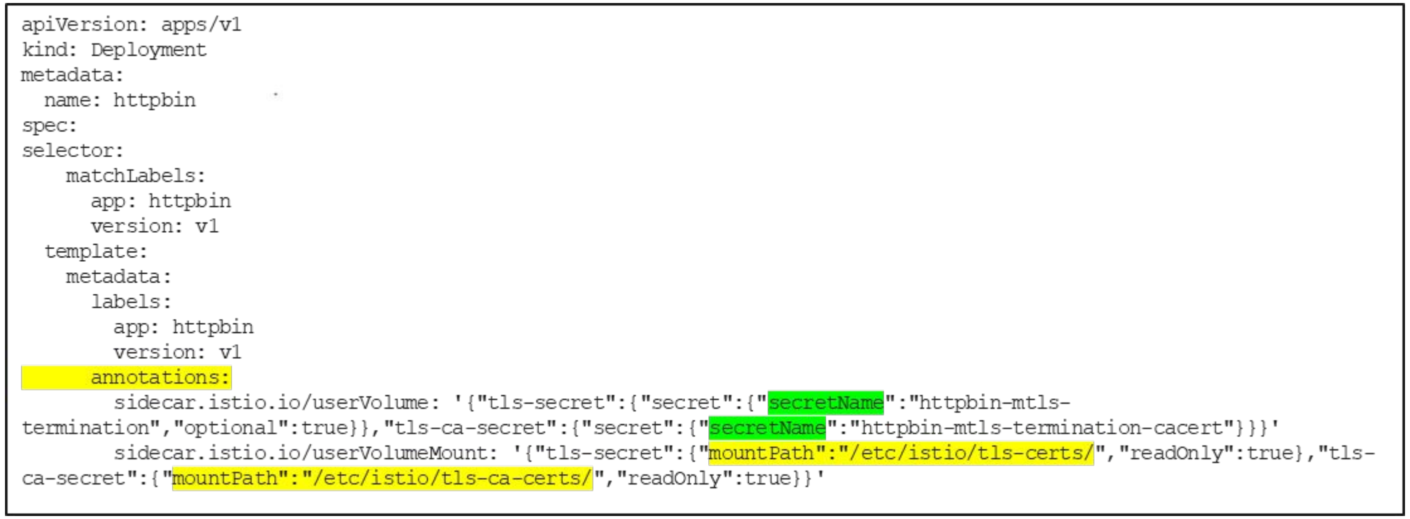
credentialName	string	The name of the secret that holds the TLS certs for the client including the CA certificates. This secret must exist in the namespace of the proxy using the certificates. An Opaque secret should contain the following keys and values: key: <privateKey>, cert: <clientCert>, cacert: <CACertificate>. Here CACertificate is used to verify the server certificate. For mutual TLS, cacert: <CACertificate> can be provided in the same secret or a separate secret named <secret>-cacert. A TLS secret for client certificates with an additional ca.crt key for CA certificates is also supported. Only one of client certificates and CA certificate or credentialName can be specified. NOTE: This field is applicable at sidecars only if DestinationRule has a workloadSelector specified. Otherwise the field will be applicable only at gateways, and sidecars will continue to use the certificate paths.	No



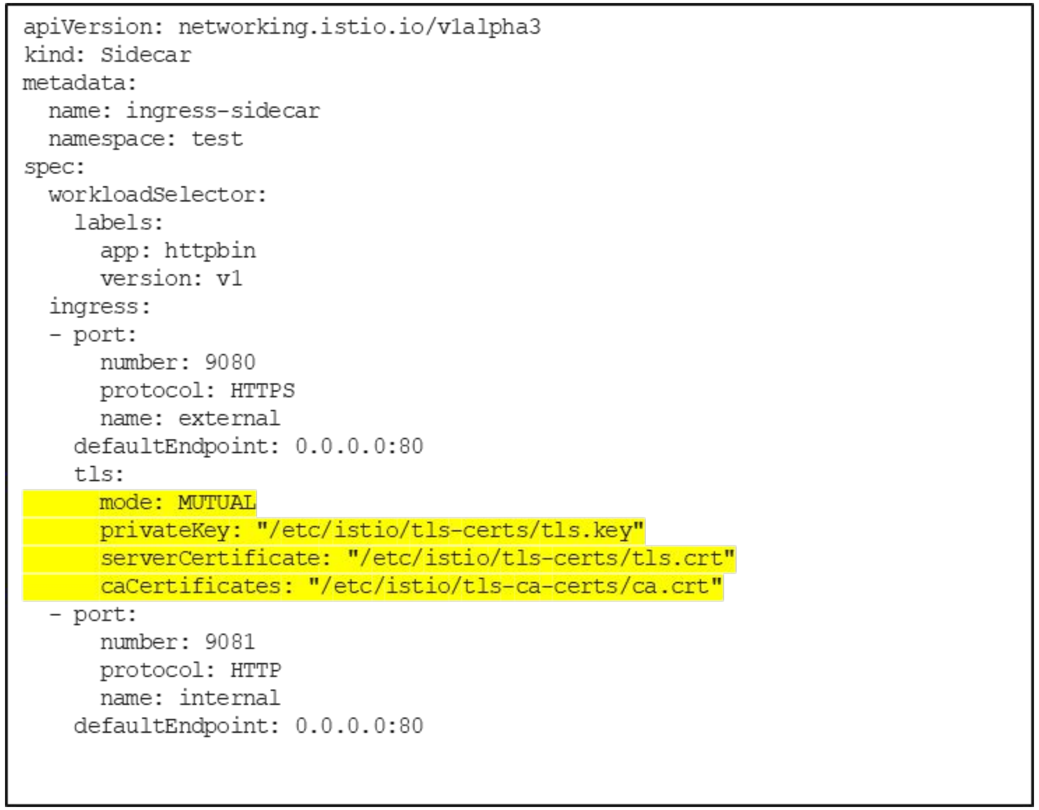
```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: originate-mtls-for-nginx
spec:
  host: my-nginx.mesh-external.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: ROUND_ROBIN
    portLevelSettings:
    - port:
        number: 443
      tls:
        mode: MUTUAL
        credentialName: client-credential # this must match the secret
created earlier to hold client certs
        sni: my-nginx.mesh-external.svc.cluster.local
```

Technical Details

- Typical use case is PM (Istio proposal for Prometheus cert handling)
- Certificates are handled at the sidecar proxy
- Ingress TLS is configured at the Istio Sidecar CR
- Configuration example from Istio
- Credential name cannot be used, Certificates need to be mounted by using annotations in application pod



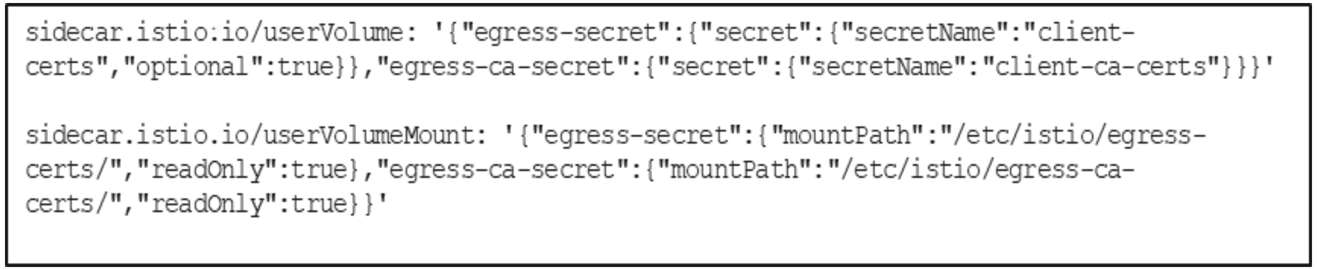
```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: httpbin
spec:
selector:
  matchLabels:
   app: httpbin
   version: v1
 template:
  metadata:
   labels:
    app: httpbin
    version: v1
   annotations:
    sidecar.istio.io/userVolume: '{"tls-secret":{"secret":{"secretName":"httpbin-mtls-
termination","optional":true}},"tls-ca-secret":{"secret":{"secretName":"httpbin-mtls-termination-cacert"}}}'
    sidecar.istio.io/userVolumeMount: '{"tls-secret":{"mountPath":"/etc/istio/tls-certs/","readOnly":true},"tls-
ca-secret":{"mountPath":"/etc/istio/tls-ca-certs/","readOnly":true}}'
```



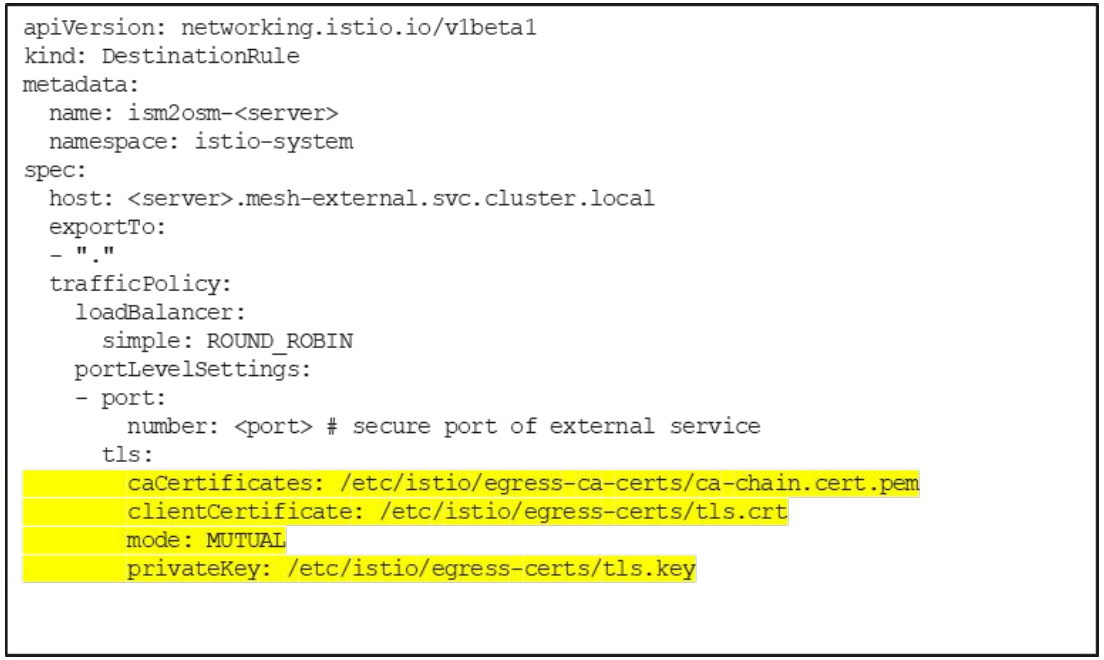
```
apiVersion: networking.istio.io/v1alpha3
kind: Sidecar
metadata:
 name: ingress-sidecar
 namespace: test
spec:
 workloadSelector:
  labels:
   app: httpbin
   version: v1
 ingress:
 - port:
   number: 9080
   protocol: HTTPS
   name: external
  defaultEndpoint: 0.0.0.0:80
  tls:
   mode: MUTUAL
   privateKey: "/etc/istio/tls-certs/tls.key"
   serverCertificate: "/etc/istio/tls-certs/tls.crt"
   caCertificates: "/etc/istio/tls-ca-certs/ca.crt"
 - port:
   number: 9081
   protocol: HTTP
   name: internal
  defaultEndpoint: 0.0.0.0:80
```

Technical Details

- Certificates are handled at Egress Pods
- Egress TLS is configured at the DestinationRule CR
- Configuration example from Istio
- Annotations are used to mount certificates



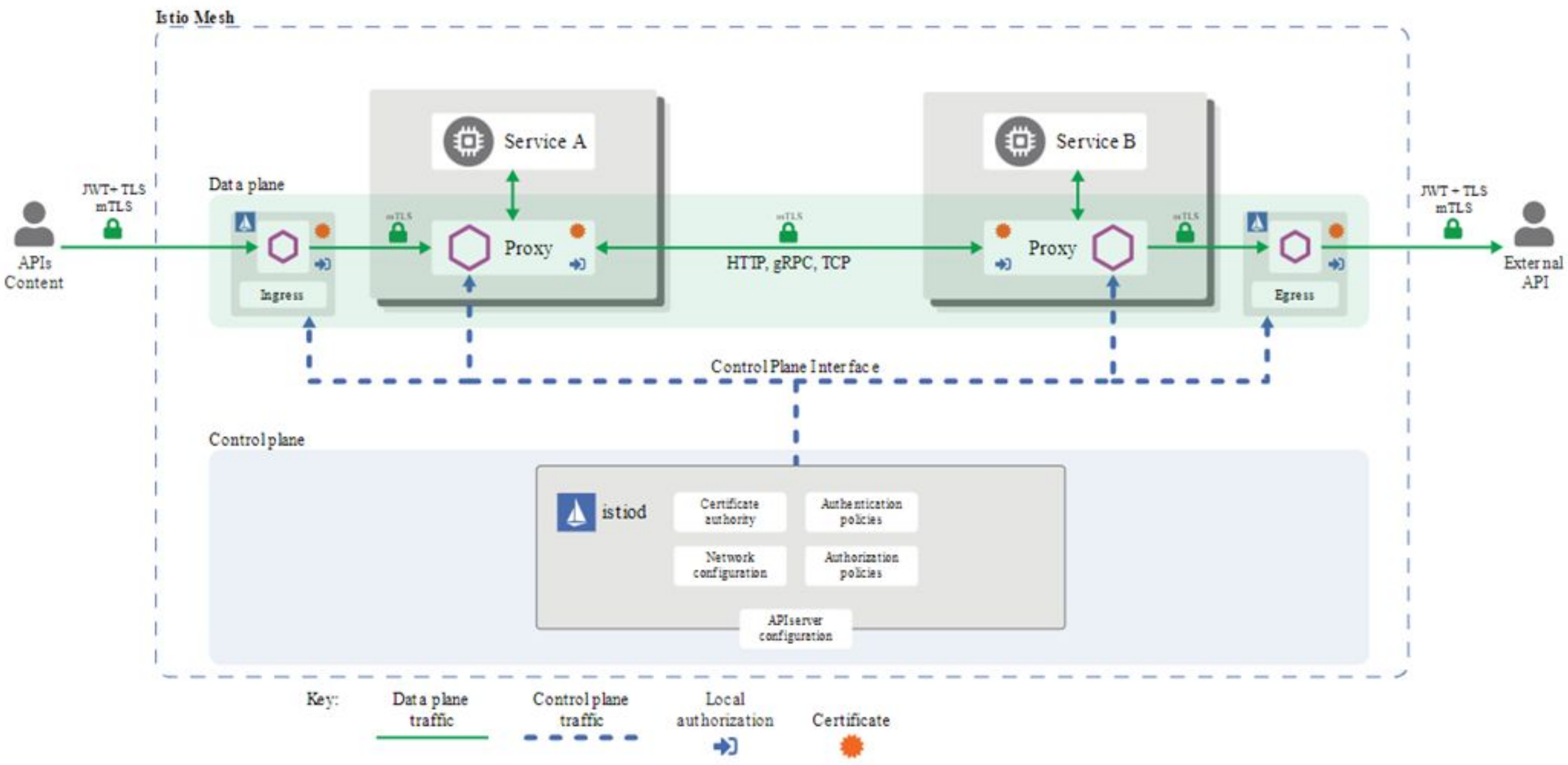
```
sidecar.istio:io/userVolume: '{"egress-secret":{"secret":{"secretName":"client-
certs","optional":true}},"egress-ca-secret":{"secret":{"secretName":"client-ca-certs"}}}'

sidecar.istio.io/userVolumeMount: '{"egress-secret":{"mountPath":"/etc/istio/egress-
certs/","readOnly":true},"egress-ca-secret":{"mountPath":"/etc/istio/egress-ca-
certs/","readOnly":true}}'
```

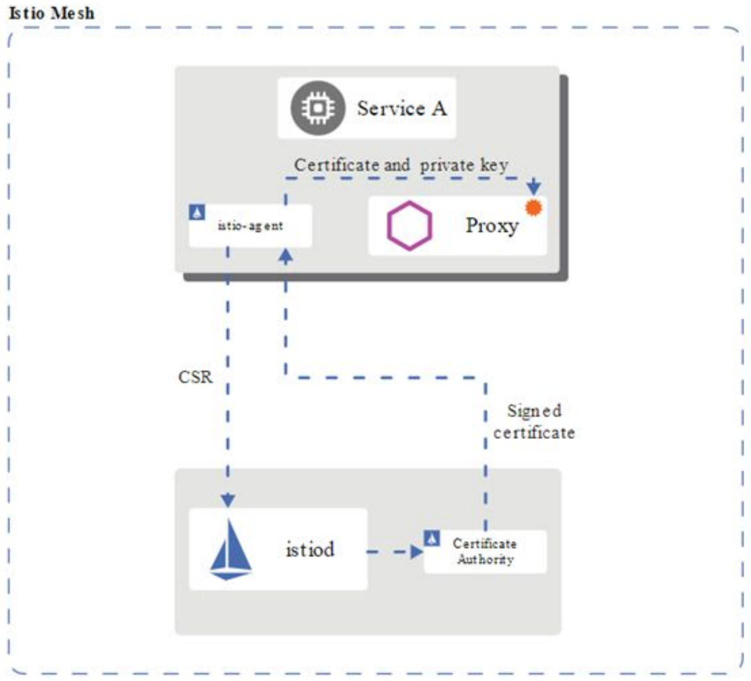


```
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: ism2osm-<server>
  namespace: istio-system
spec:
  host: <server>.mesh-external.svc.cluster.local
  exportTo:
  - "."
  trafficPolicy:
    loadBalancer:
      simple: ROUND_ROBIN
    portLevelSettings:
    - port:
        number: <port> # secure port of external service
      tls:
        caCertificates: /etc/istio/egress-ca-certs/ca-chain.cert.pem
        clientCertificate: /etc/istio/egress-certs/tls.crt
        mode: MUTUAL
        privateKey: /etc/istio/egress-certs/tls.key
```

Istio Security Architecture

- External certificates to be used at Ingress / Egress Gateway, these certificates are typically stored in secrets
- Istio internal certificates based on SPIFFE handled by default by Istio CA (Certificate Authority)
- Identity provisioning flow shown in the picture below, certificate + key information only kept in memory of the sidecar proxy, no secrets required

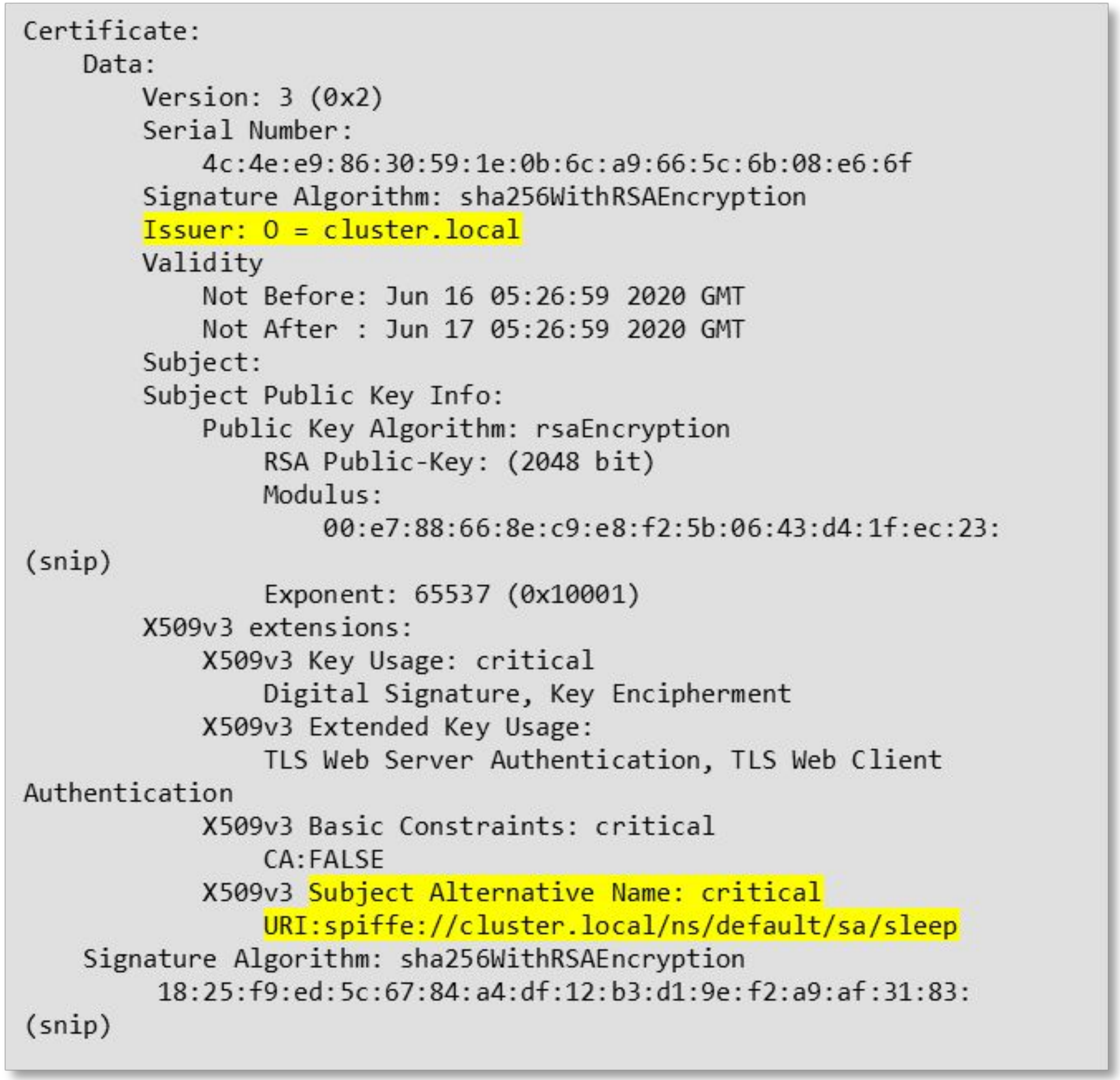




Identity Provisioning Workflow

Istio certificates and SPIFFE

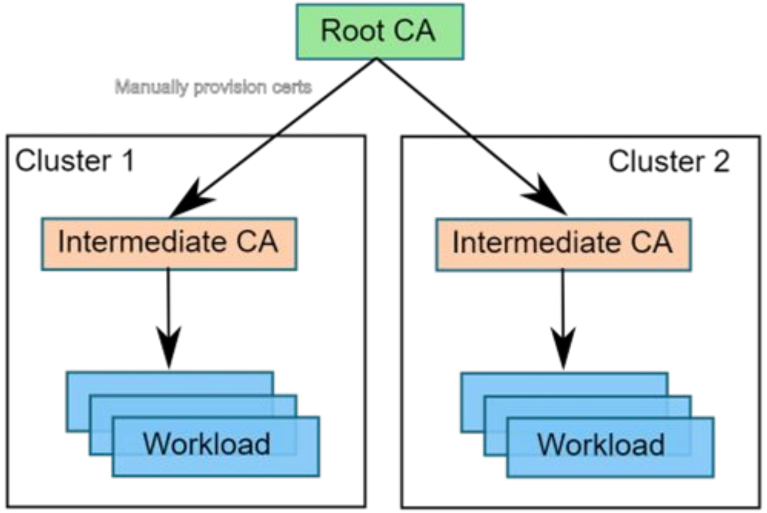
- Service Mesh certificate are regular X.509 certificates
- They carry an identity in SPIFFE format (link)
- –SubjectName field is set to *spiffe://<domain>/ns/<namespace>/sa/<serviceaccount>*
- –where:
  - *<domain>* is configurable at installation time, and defaults to "cluster.local"
  - *<namespace>* indicates the namespace the Pod belongs to
  - *<serviceaccount>* indicates the K8s ServiceAccount the Pod runs under.
- SPIFFE identities are the foundation for Istio authorization framework (link)
  - Istio can create a SPIFFE certificate and key for K8s ServiceAccounts.
    - All Pods running under the same ServiceAccount have a client certificate with the same Subject Name.
  - Access authorization to services is done by defining AuthorizationPolicy objects
  - Authorization is optional, default authorization authorizes all accesses

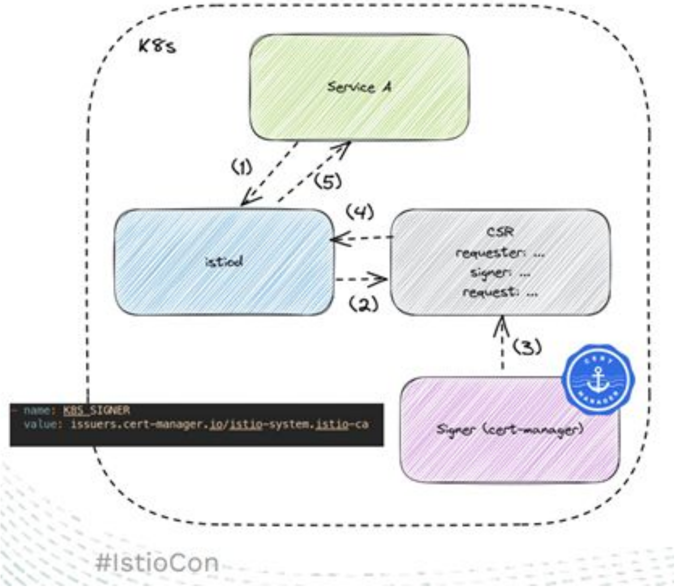


```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            4c:4e:e9:86:30:59:1e:0b:6c:a9:66:5c:6b:08:e6:6f
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = cluster.local
        Validity
            Not Before: Jun 16 05:26:59 2020 GMT
            Not After : Jun 17 05:26:59 2020 GMT
        Subject:
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:e7:88:66:8e:c9:e8:f2:5b:06:43:d4:1f:ec:23:
(snip)
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client
Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Subject Alternative Name: critical
                URI:spiffe://cluster.local/ns/default/sa/sleep
    Signature Algorithm: sha256WithRSAEncryption
         18:25:f9:ed:5c:67:84:a4:df:12:b3:d1:9e:f2:a9:af:31:83:
(snip)
```

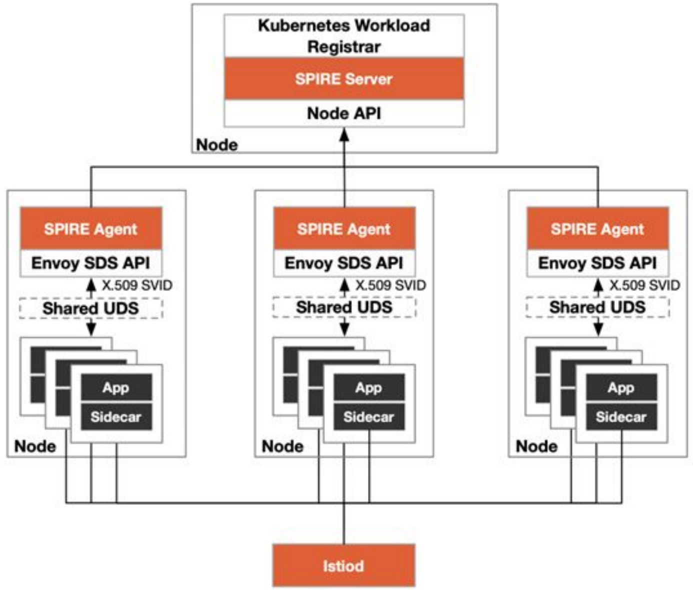
CA certificate handling in Istio

- By default, Istio will create self signed root CA cert (own internal PKI), 10 years lifetime
- Root CA cert is stored in secret incl. private key , root CA cert distributed via configmap
- Possible to plugin intermediate CA cert to integrate with another CA (like from AWS)
- Possible to use custom CA by using K8S CSR API (e.g., using cert-manager, avoids storing priv key)
- Possible to use SPIRE as CA, see blog post from Tetrate (requires DaemonSet)
- Possible to integrate directly with Vault (blog post from Tetrate)



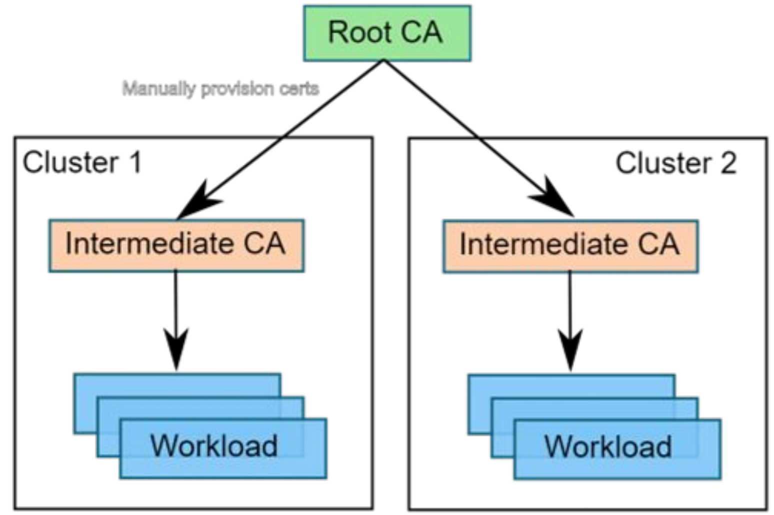


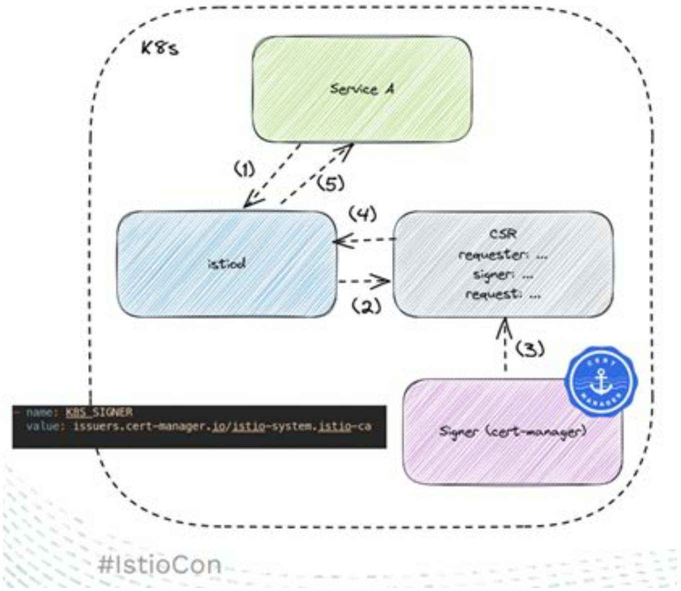
```
- name: K8S_SIGNER
  value: issuers.cert-manager.io/istio-system.istio-ca
```



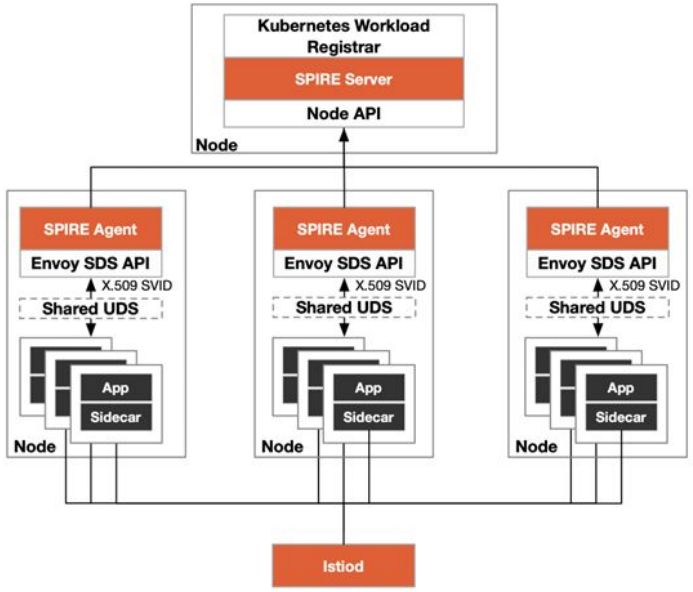
CA certificate handling in Istio

- By default, Istio will create self signed root CA cert (own internal PKI), 10 years lifetime
- Root CA cert is stored in secret incl. private key , root CA cert distributed via configmap
- Possible to plugin intermediate CA cert to integrate with another CA (like from SIP-TLS or AWS)
- Possible to use custom CA by using K8S CSR API (e.g., using cert-manager, avoids storing priv key)
- Possible to use SPIRE as CA, see blog post from Tetrate (requires DaemonSet)
- Possible to integrate directly with Vault (blog post from Tetrate)





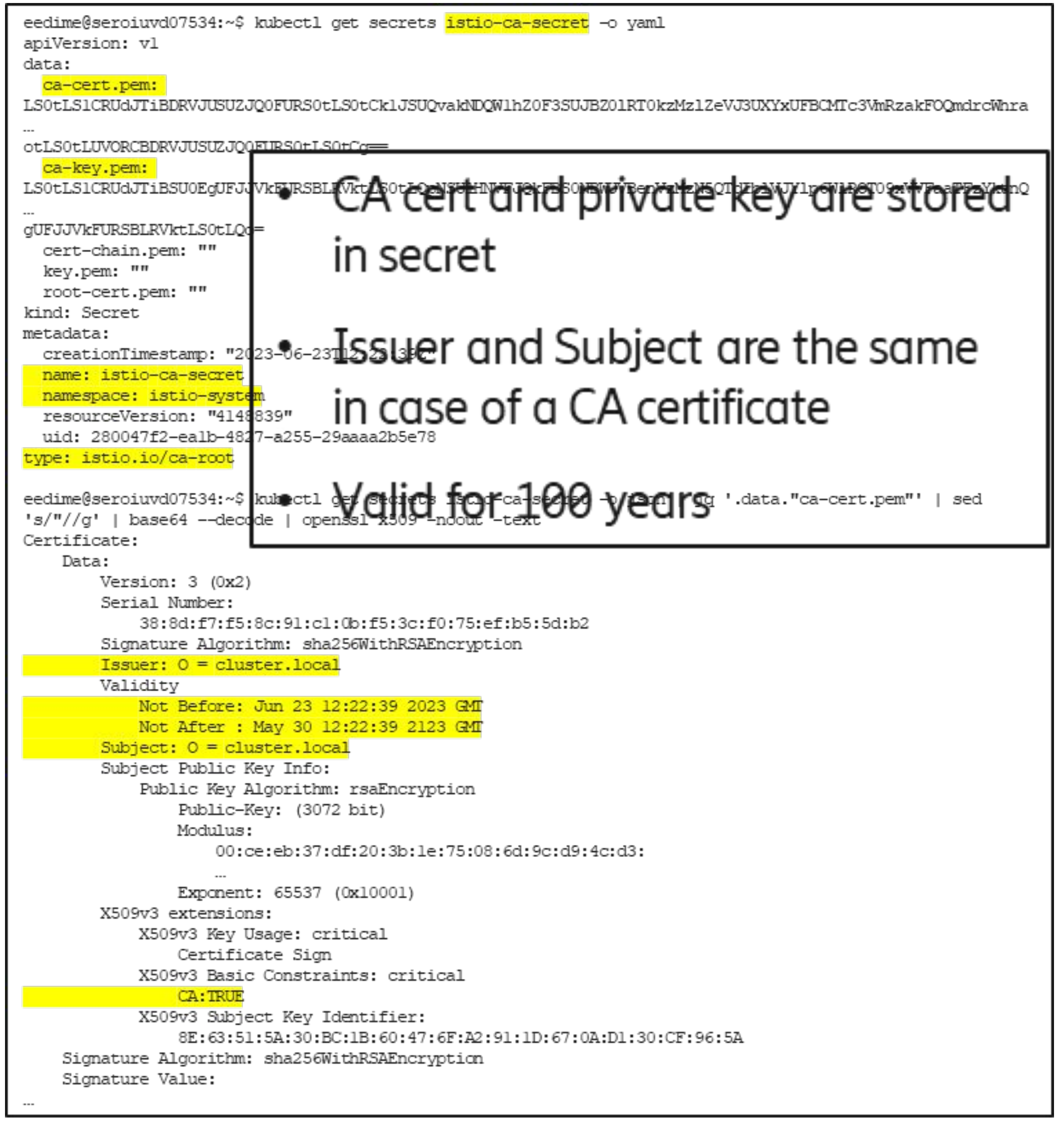
```
- name: K8S_SIGNER
  value: issuers.cert-manager.io/istio-system.istio-ca
```



Where to find which certificate

KubeCon | CloudNativeCon
North America 2023

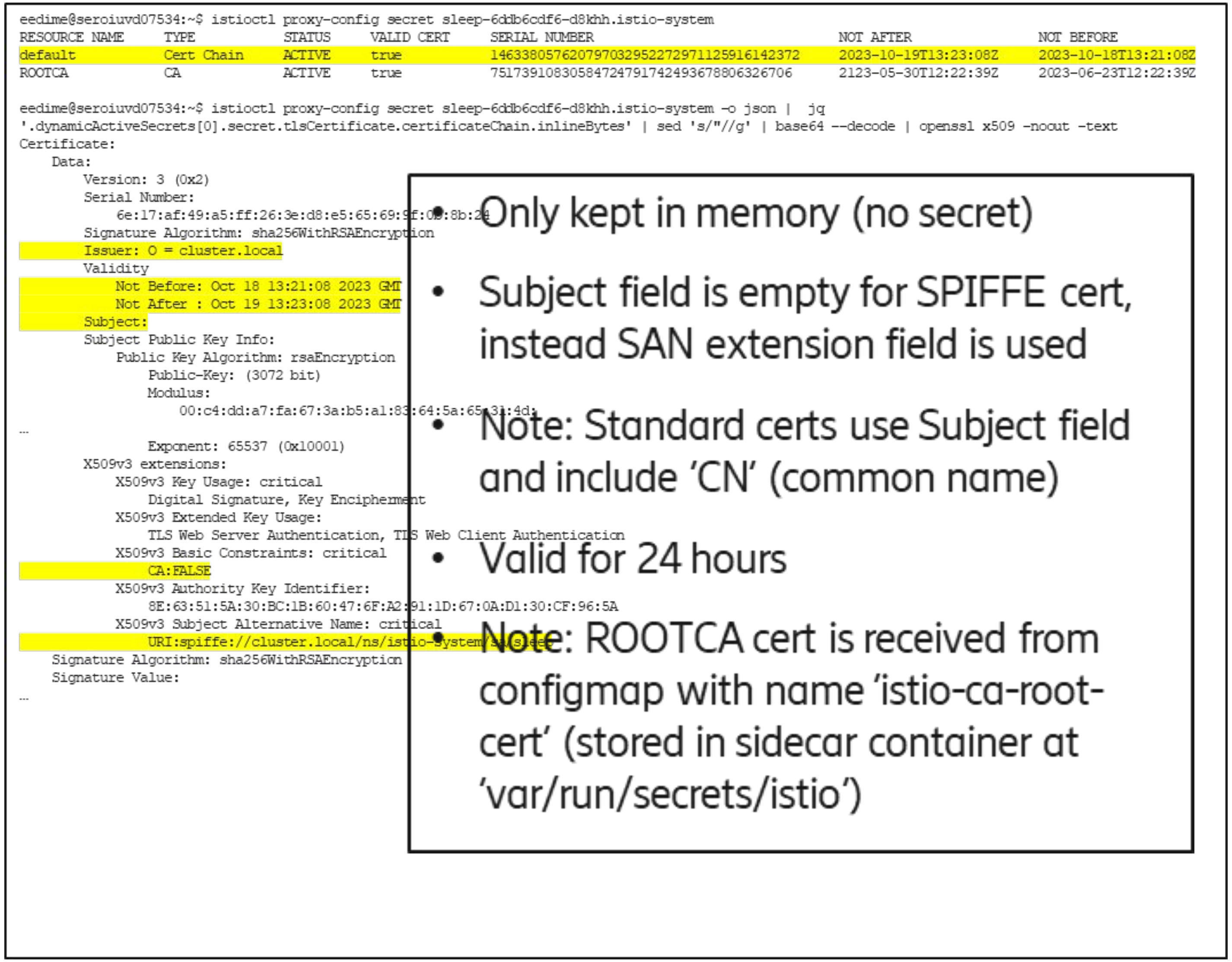
CA Certificate



```
eedime@seroiuvd07534:~$ kubectl get secrets istio-ca-secret -o yaml
apiVersion: v1
data:
  ca-cert.pem:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUQvakNDQW1hZ0F3SUJBZ01RT0kzMzlZeVJ3UXYxUFBCMTc3VmRzakFOQmdrcWhra
...
otLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
  ca-key.pem:
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQ
...
gUFJJVkFURSBLRVktLS0tLQo=
  cert-chain.pem: ""
  key.pem: ""
  root-cert.pem: ""
kind: Secret
metadata:
  creationTimestamp: "2023-06-23T12:22:39Z"
  name: istio-ca-secret
  namespace: istio-system
  resourceVersion: "4148839"
  uid: 280047f2-ea1b-4827-a255-29aaaa2b5e78
type: istio.io/ca-root

eedime@seroiuvd07534:~$ kubectl get secrets istio-ca-secret -o json | jq '.data."ca-cert.pem"' | sed
's/"//g' | base64 --decode | openssl x509 -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            38:8d:f7:f5:8c:91:c1:0b:f5:3c:f0:75:ef:b5:5d:b2
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = cluster.local
        Validity
            Not Before: Jun 23 12:22:39 2023 GMT
            Not After : May 30 12:22:39 2123 GMT
        Subject: O = cluster.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (3072 bit)
                Modulus:
                    00:ce:eb:37:df:20:3b:1e:75:08:6d:9c:d9:4c:d3:
                    ...
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                8E:63:51:5A:30:BC:1B:60:47:6F:A2:91:1D:67:0A:D1:30:CF:96:5A
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
...
```

- CA cert and private key are stored in secret
- Issuer and Subject are the same in case of a CA certificate
- Valid for 100 years

Service specific SPIFFE certificate in sidecar



```
eedime@seroiuvd07534:~$ istioctl proxy-config secret sleep-6ddb6cdf6-d8khh.istio-system
RESOURCE NAME     TYPE           STATUS     VALID CERT     SERIAL NUMBER                                  NOT AFTER                NOT BEFORE
default           Cert Chain     ACTIVE     true           146338057620797032952272971125916142372        2023-10-19T13:23:08Z     2023-10-18T13:21:08Z
ROOTCA            CA             ACTIVE     true           75173910830584724791742493678806326706         2123-05-30T12:22:39Z     2023-06-23T12:22:39Z

eedime@seroiuvd07534:~$ istioctl proxy-config secret sleep-6ddb6cdf6-d8khh.istio-system -o json |  jq
'.dynamicActiveSecrets[0].secret.tlsCertificate.certificateChain.inlineBytes' | sed 's/"//g' | base64 --decode | openssl x509 -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            6e:17:af:49:a5:ff:26:3e:d8:e5:65:69:9f:0b:8b:24
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = cluster.local
        Validity
            Not Before: Oct 18 13:21:08 2023 GMT
            Not After : Oct 19 13:23:08 2023 GMT
        Subject:
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (3072 bit)
                Modulus:
                    00:c4:dd:a7:fa:67:3a:b5:a1:83:64:5a:65:31:4d:
...
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Authority Key Identifier:
                8E:63:51:5A:30:BC:1B:60:47:6F:A2:91:1D:67:0A:D1:30:CF:96:5A
            X509v3 Subject Alternative Name: critical
                URI:spiffe://cluster.local/ns/istio-system/sa/sleep
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
...
```

- Only kept in memory (no secret)
- Subject field is empty for SPIFFE cert, instead SAN extension field is used
- Note: Standard certs use Subject field and include 'CN' (common name)
- Valid for 24 hours
- Note: ROOTCA cert is received from configmap with name 'istio-ca-root-cert' (stored in sidecar container at 'var/run/secrets/istio')