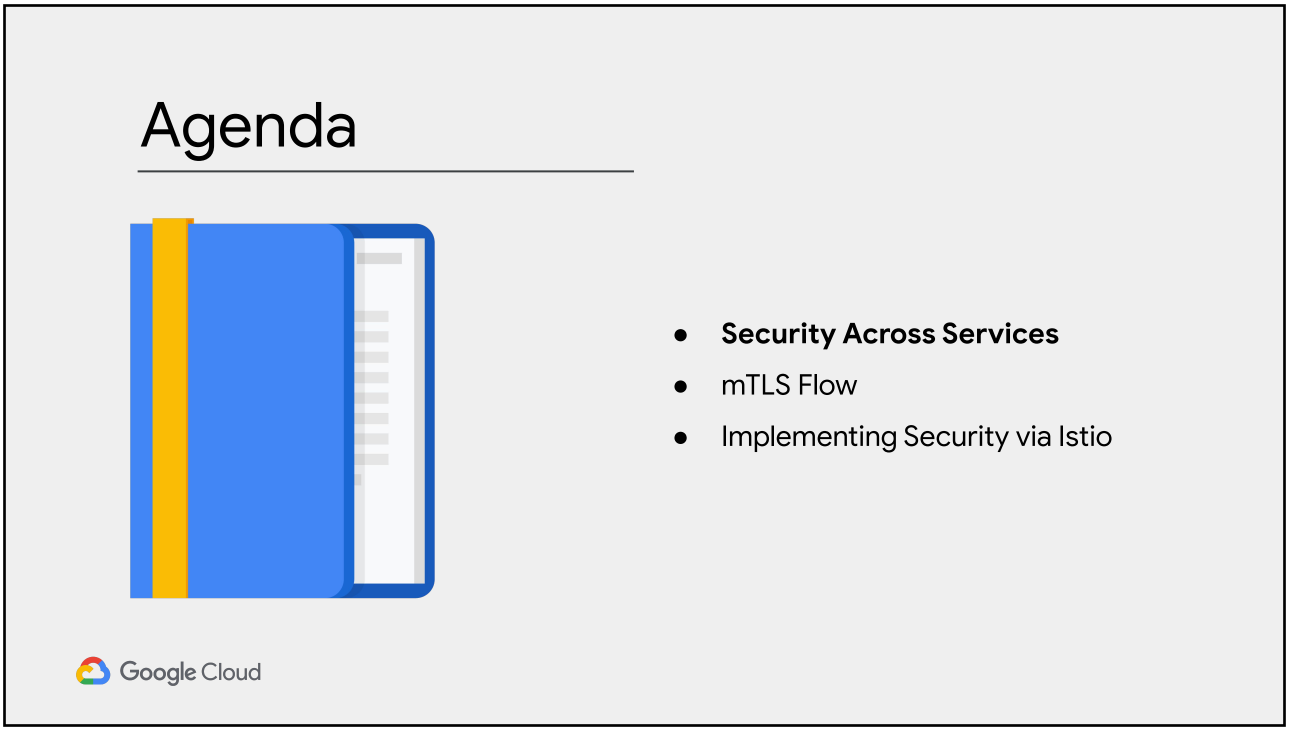
Google Cloud

Architecting Hybrid Infrastructure with Anthos

Securing your Services with Service Mesh



Agenda

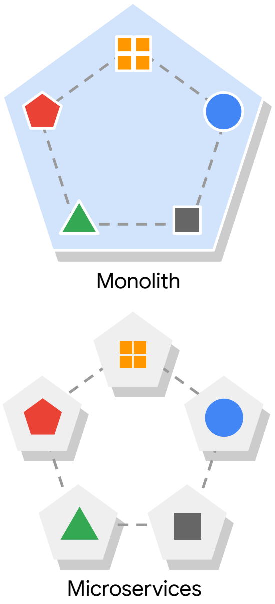
- **Security Across Services**
- mTLS Flow
- Implementing Security via Istio

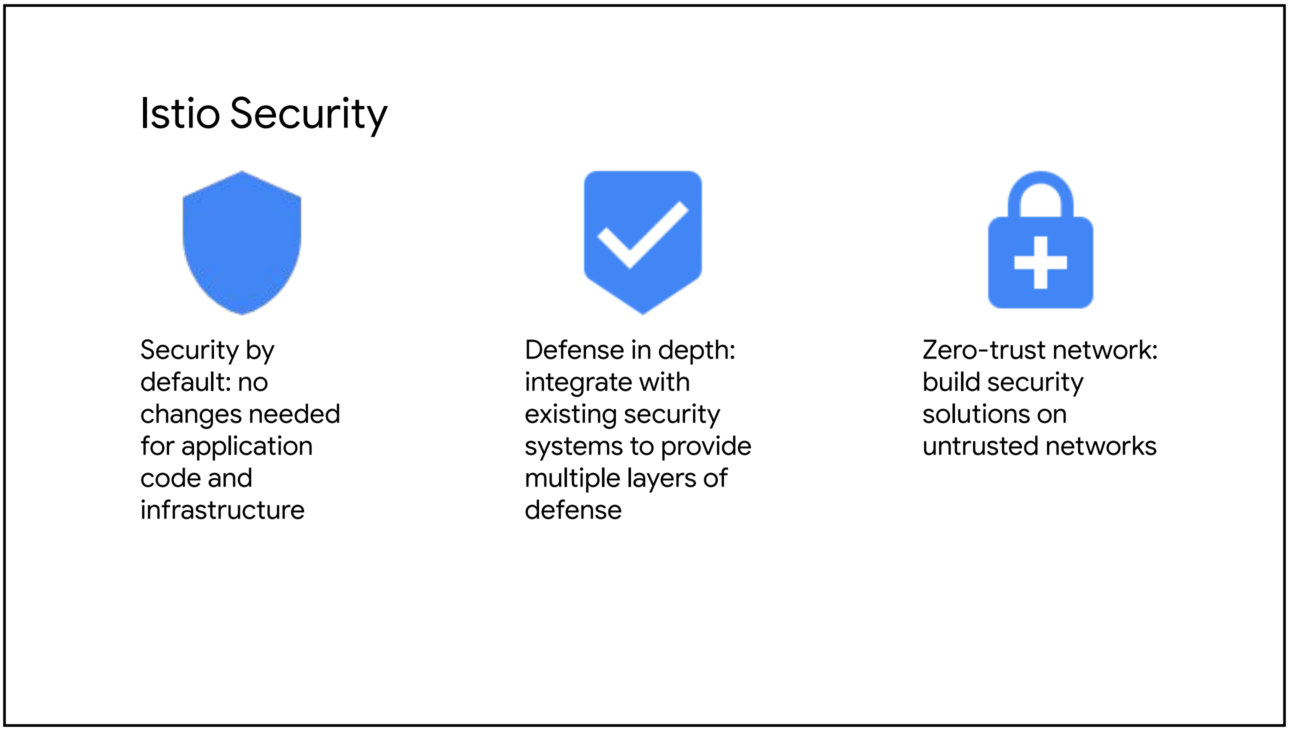
Monolith to Microservices

To defend against the man-in-the-middle attack, they need traffic encryption

To provide flexible service access control, they need mutual TLS and fine-grained access policies

To audit who did what at what time, they need auditing tools





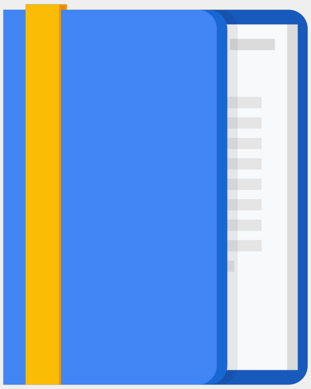
Istio Security

Security by default: no changes needed for application code and infrastructure

Defense in depth: integrate with existing security systems to provide multiple layers of defense

Zero-trust network: build security solutions on untrusted networks

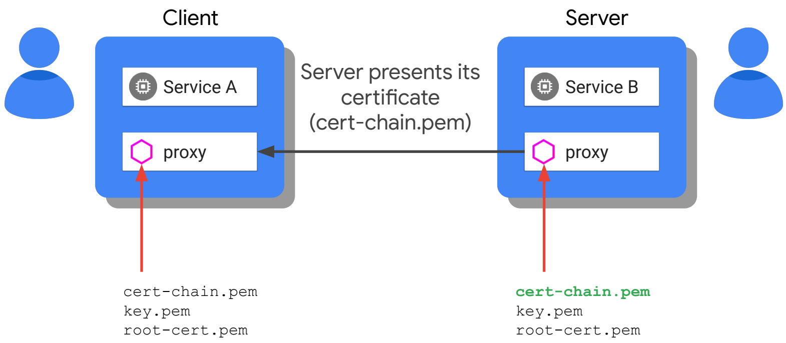
Agenda



- **Security Across Services**
- mTLS Flow
- Implementing Security via Istio

Google Cloud

mTLS (Mutual TLS) - Flow



Client

Server

Service A

Service B

proxy

proxy

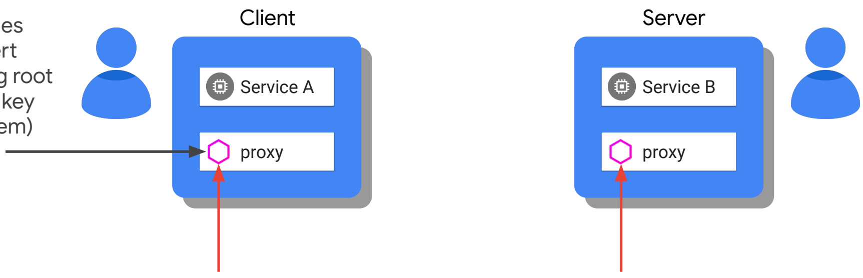
Server presents its certificate (cert-chain.pem)

```
cert-chain.pem
key.pem
root-cert.pem
```

```
cert-chain.pem
key.pem
root-cert.pem
```

mTLS (Mutual TLS) - Flow

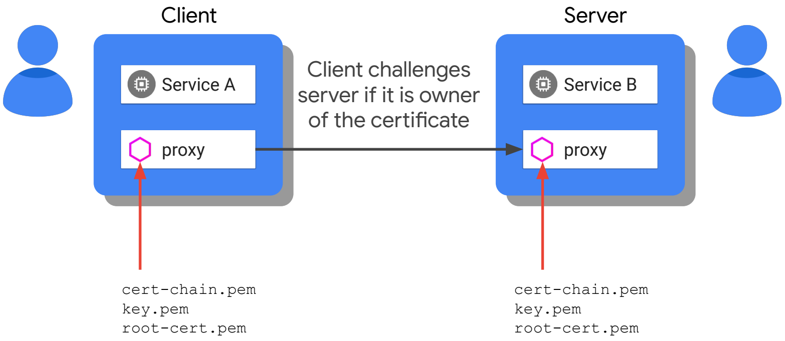
Client verifies server's cert signature using root cert's public key (root-cert.pem)



cert-chain.pem
key.pem
root-cert.pem

cert-chain.pem
key.pem
root-cert.pem

mTLS (Mutual TLS) - Flow

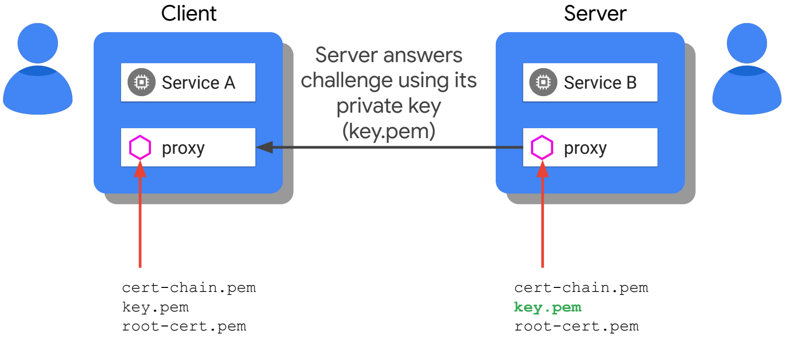


Client challenges server if it is owner of the certificate

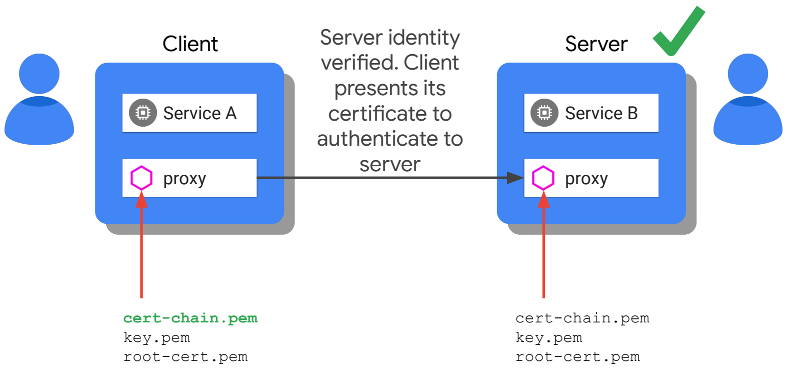
```
cert-chain.pem
key.pem
root-cert.pem
```

```
cert-chain.pem
key.pem
root-cert.pem
```

mTLS (Mutual TLS) - Flow



mTLS (Mutual TLS) - Flow



Client

Service A

proxy

Server identity verified. Client presents its certificate to authenticate to server

Server

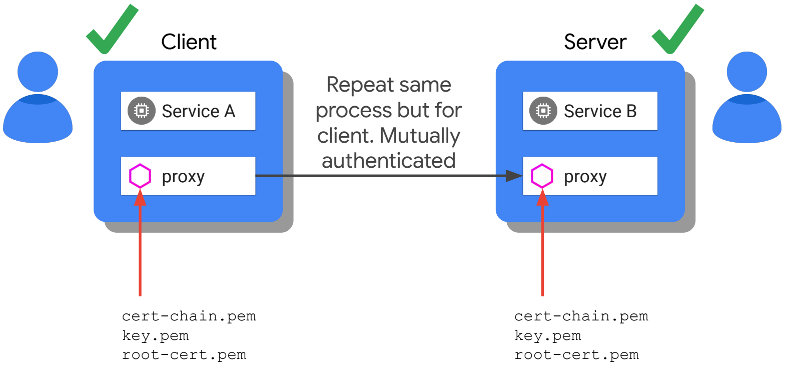
Service B

proxy

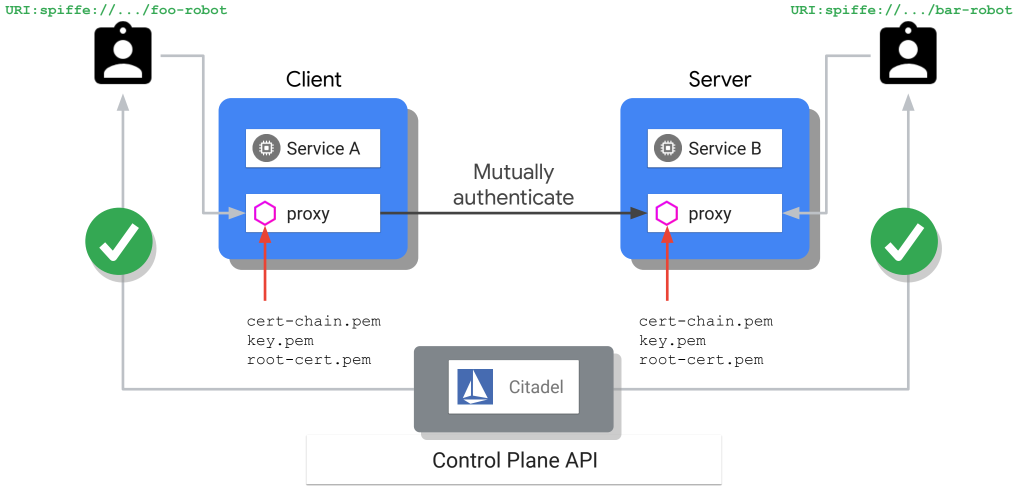
```
cert-chain.pem
key.pem
root-cert.pem
```

```
cert-chain.pem
key.pem
root-cert.pem
```

mTLS (Mutual TLS) - Flow



mTLS (Mutual TLS)



URI:spiffe://.../foo-robot

URI:spiffe://.../bar-robot

Client

Server

Service A

proxy

Mutually authenticate

Service B

proxy

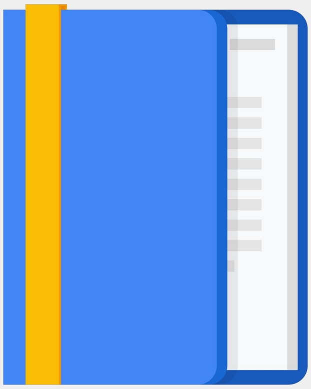
```
cert-chain.pem
key.pem
root-cert.pem
```

```
cert-chain.pem
key.pem
root-cert.pem
```

Citadel

Control Plane API

Agenda

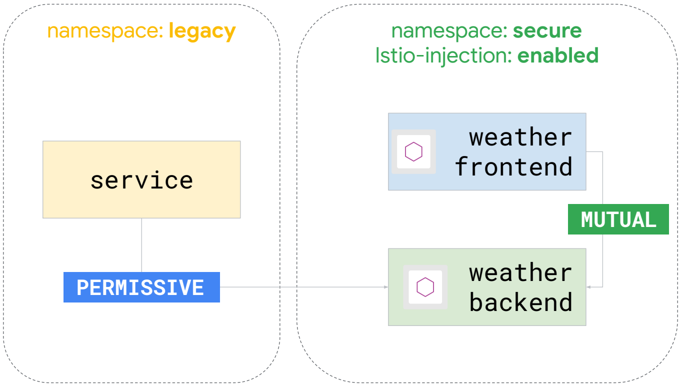


- **Security Across Services**
- mTLS Flow
- Implementing Security via Istio

Google Cloud

Apply DestinationRule with MUTUAL mode

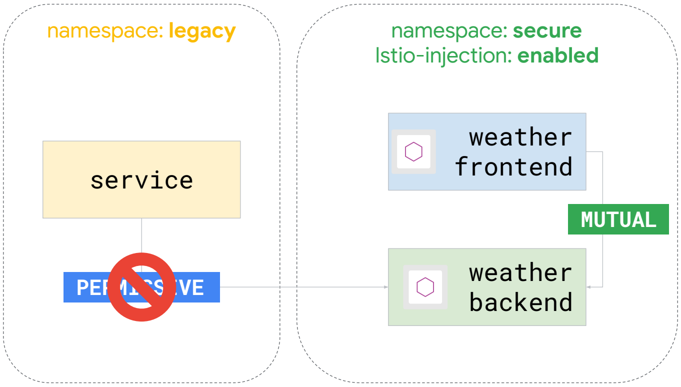
```
apiVersion: net.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: mtls-mutual
spec:
  host: adservice.secure
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```



Google Cloud

Apply Policy with STRICT mode

```yaml
apiVersion: auth.istio.io/v1alpha1
kind: Policy
metadata:
  name: mtls-backend
  namespace: secure
spec:
  targets:
  - name: adservice
  peers:
  - mtls:
      mode: STRICT
```

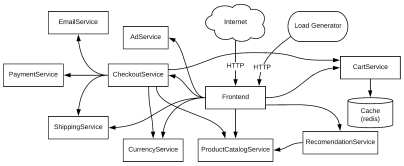


Google Cloud

Lab

Managing Policies and Security with Istio and Citadel

35 min



Objectives

- Deploy Hipster Shop, an Istio-enabled multi-service application
- Understand authentication and enable service to service authentication with mTLS
- Enable end-user JWT authentication alongside mTLS
- Understand Istio authorization and enable frontend authorization