KubeCon CloudNativeCon

OPEN SOURCE SUMMIT

China 2019

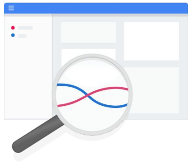
How SPIFFE helps Istio in Service Mesh Federation

Yonggang (Oliver) Liu *yonggangl@google.com*
Wencheng Lu *wlu@google.com*

What is Istio?

A service mesh. But more: An open services platform to manage service interactions across container- and VM-based workloads

Uniform observability



Operational agility

Policy driven security

What is SPIFFE?

A set of open-source standards to provide a secure production identity framework in a heterogeneous environment.

- SVID (SPIFFE Verifiable Identity Document): Standardize SPIFFE identity in X.509 certificate format
- SPIFFE APIs: A set of APIs and specs to describe how to securely **provision and federate** SPIFFE identities

Istio Identity Framework

A secure identity framework to provide strong identities for service-to-service authentication

- SVID compliant
- Support SPIFFE federation API: work in progress

Why Service Mesh Federation Matters?

Provide the interoperability between:

- Service meshes from different Orgs
- Service meshes from the same Org
- Service meshes from different Cloud and On-prem
- Service meshes from different env: k8s mesh and VM mesh

Identity Federation Challenges

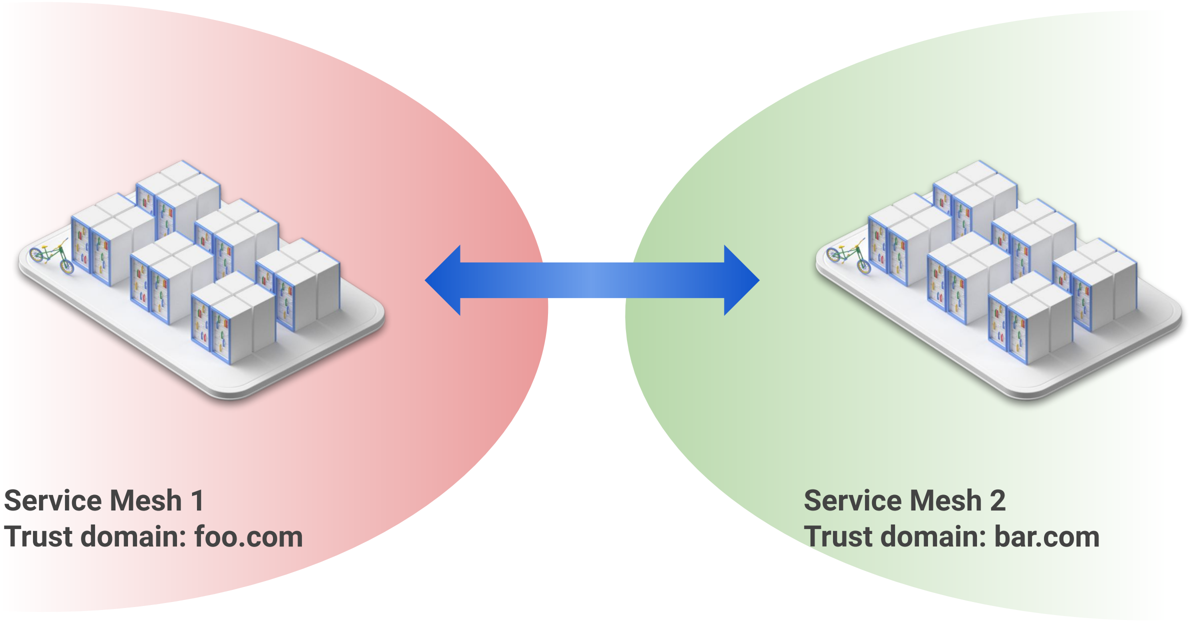
- Build identity trust between meshes
- Identity isolations

Service Mesh and Trust Domain

- In current Istio, the applications in a service mesh share ***common roots of trust*** and the same ***trust domain***.
- A trust domain could represent an individual, organization, environment or department running their own independent SPIFFE infrastructure.
- The trust domain is encoded in Istio/SPIFFE identities: *spiffe://**<trust_domain>**/ns/<K8s namespace>/sa/<K8s service account>*

Federation of Meshes

- For the applications in two meshes to authenticate, they need to verify each other’s certificates using their own trusted roots.

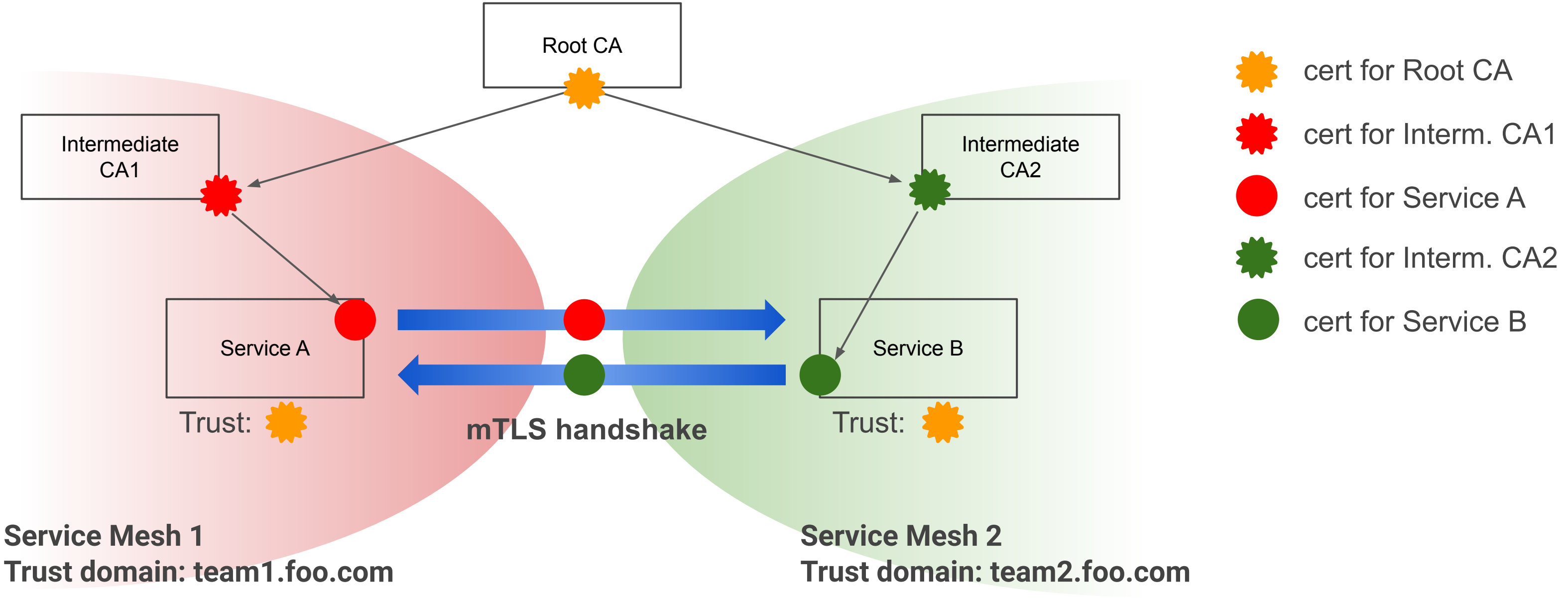


Service Mesh 1
Trust domain: foo.com

Service Mesh 2
Trust domain: bar.com

Federation within an Organization

- Common root CA
- Intermediate CA name constraints help to isolate trust domains



Root CA

Intermediate CA1

Intermediate CA2

Service A

Service B

Trust:

Trust:

mTLS handshake

Service Mesh 1
Trust domain: team1.foo.com

Service Mesh 2
Trust domain: team2.foo.com

cert for Root CA

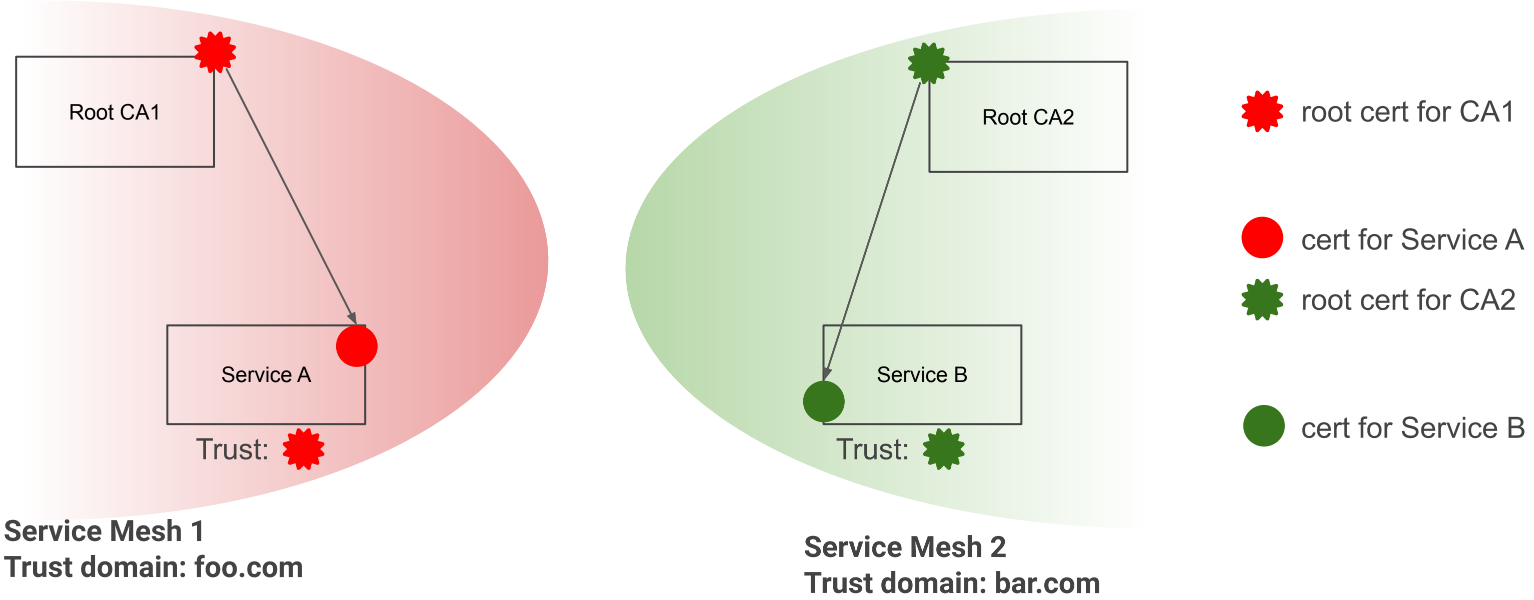
cert for Interm. CA1

cert for Service A

cert for Interm. CA2

cert for Service B

Federation across Organizations (1)



Root CA1

Service A

Trust: ✹

Root CA2

Service B

Trust: ✹

Service Mesh 1
Trust domain: foo.com

Service Mesh 2
Trust domain: bar.com

✹ root cert for CA1

● cert for Service A

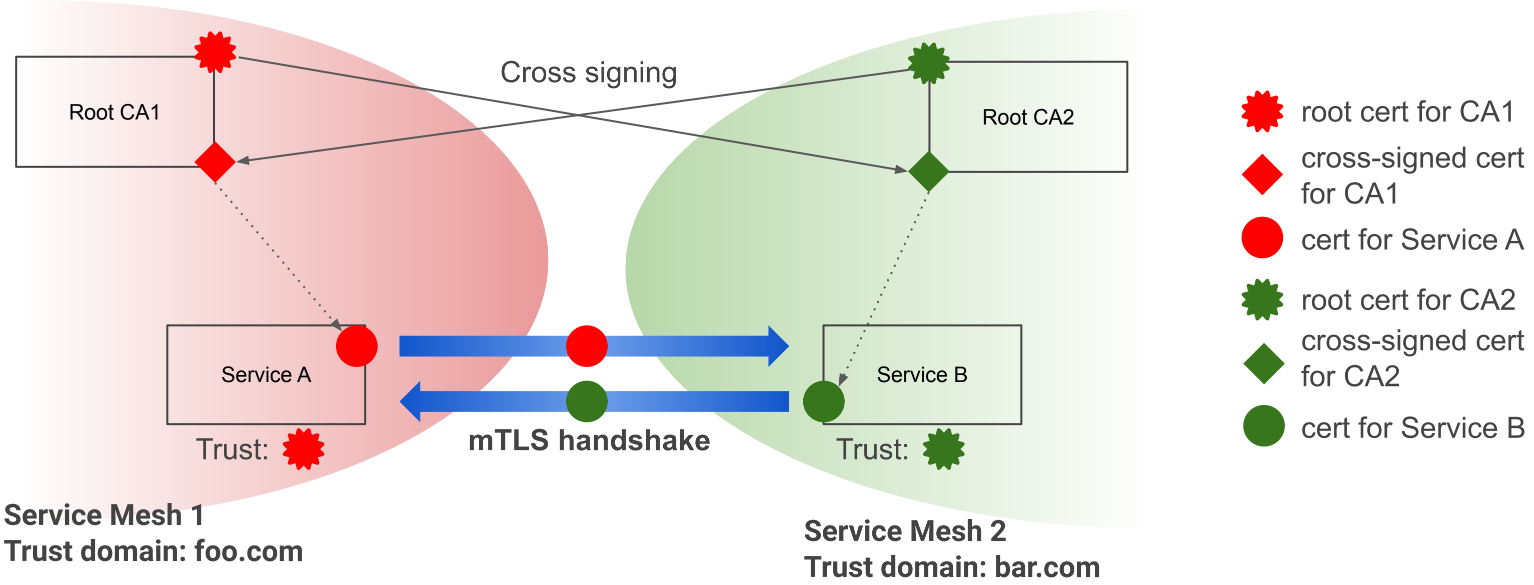
✹ root cert for CA2

● cert for Service B

Federation across Organizations (1)

Root CA Cross Certification

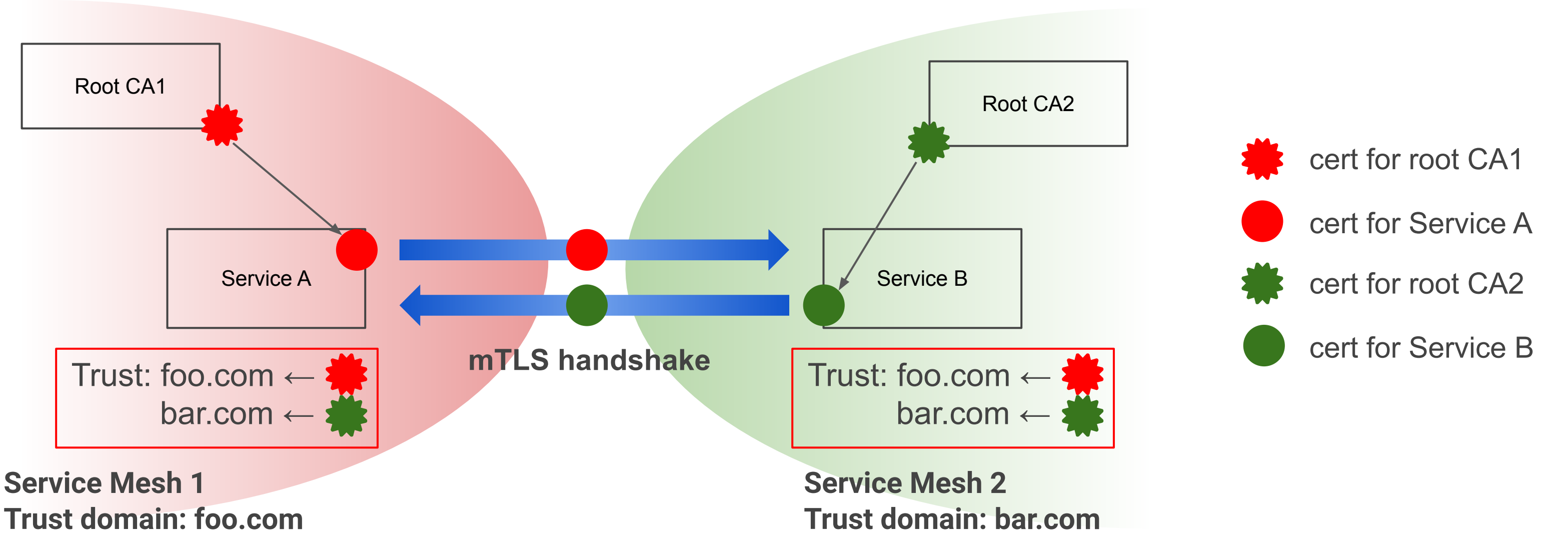
- High complexity and not scalable. O(N^2) cross-signings for N trust domains



Federation across Organizations (2)

SPIFFE Trust Bundle (recommended)

- Automated root of trust exchange
- Authentication using the root certs corresponding to the peer’s trust domain



SPIFFE Trust Bundle

- A SPIFFE trust bundle is an *RFC 7517 compliant JWK set* containing a trust domain's cryptographic keys, for the validation of certificates issued in that trust domain.

SPIFFE Trust Bundle Example

```
{
    "keys": [
        {
            "use": "x509-svid",
            "kty": "EC",
            "crv": "P-256",
            "x": "fK-wKTnKL7KFLM27lqq5DC-bxrVaH6rDV-IcCSEOeL4",
            "y": "wq-g3TQWxYlV51TCPH030yXsRxvujD4hUUaIQrXk4KI",
            "x5c": [

"MIIBKjCB0aADAgECAgEBMAoGCCqGSM49BAMCMAAwIhgPMDAwMTAxMDEwMDAwMDBaGA85OTk5MTIzMTIzNTk1OVowADBZMBMGByqGSM49AgEGCCqGSM49Aw
EHA0IABHyvsCk5yi+yhSzNu5aquQwvm8a1Wh+qw1fiHAkhDni+wq+g3TQWxYlV51TCPH030yXsRxvujD4hUUaIQrXk4KKjODA2MA8GA1UdEwEB/wQFMAMBA
f8wIwYDVR0RAQH/BBkwF4YVc3BpZmZlOi8vZG9tYWluMS50ZXN0MAoGCCqGSM49BAMCA0gAMEUCIA2dO09Xmakw2ekuHKWC4hBhCkpr5qY4bI8YUcXfxg/1
AiEA67kMyH7bQnr7OVLUrL+b9ylAdZglS5kKnYigmwDh+/U="
            ]
        }
    ],
    "spiffe_refresh_hint": 600
}
```

Federation with SPIFFE Trust Bundle (1)

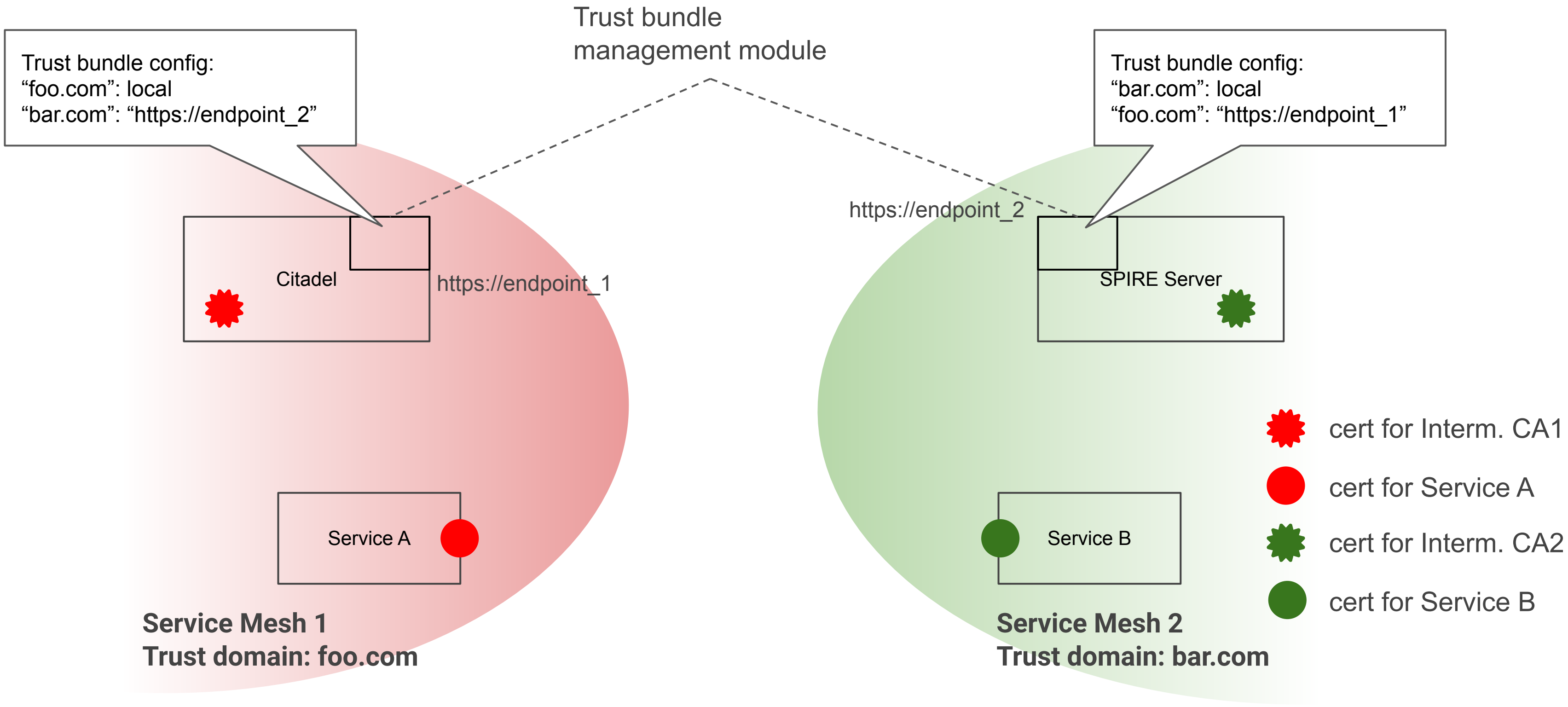
Publishing the trust bundle

- **HTTPS** endpoint, TLS cert based on WebPKI or SPIFFE

Consuming the trust bundle

- Admin configures the **<trust_domain, endpoint>** mapping
- Istio authenticates the HTTPS endpoint and retrieves the bundle
- <trust_domain, bundle> tuples are propagated to each workload and used in cert verification

Federation with SPIFFE Trust Bundle (2)



Trust bundle management module

Trust bundle config:
“foo.com”: local
“bar.com”: “https://endpoint_2”

Citadel

https://endpoint_1

Service A

Service Mesh 1
Trust domain: foo.com

Trust bundle config:
“bar.com”: local
“foo.com”: “https://endpoint_1”

https://endpoint_2

SPIRE Server

Service B

Service Mesh 2
Trust domain: bar.com

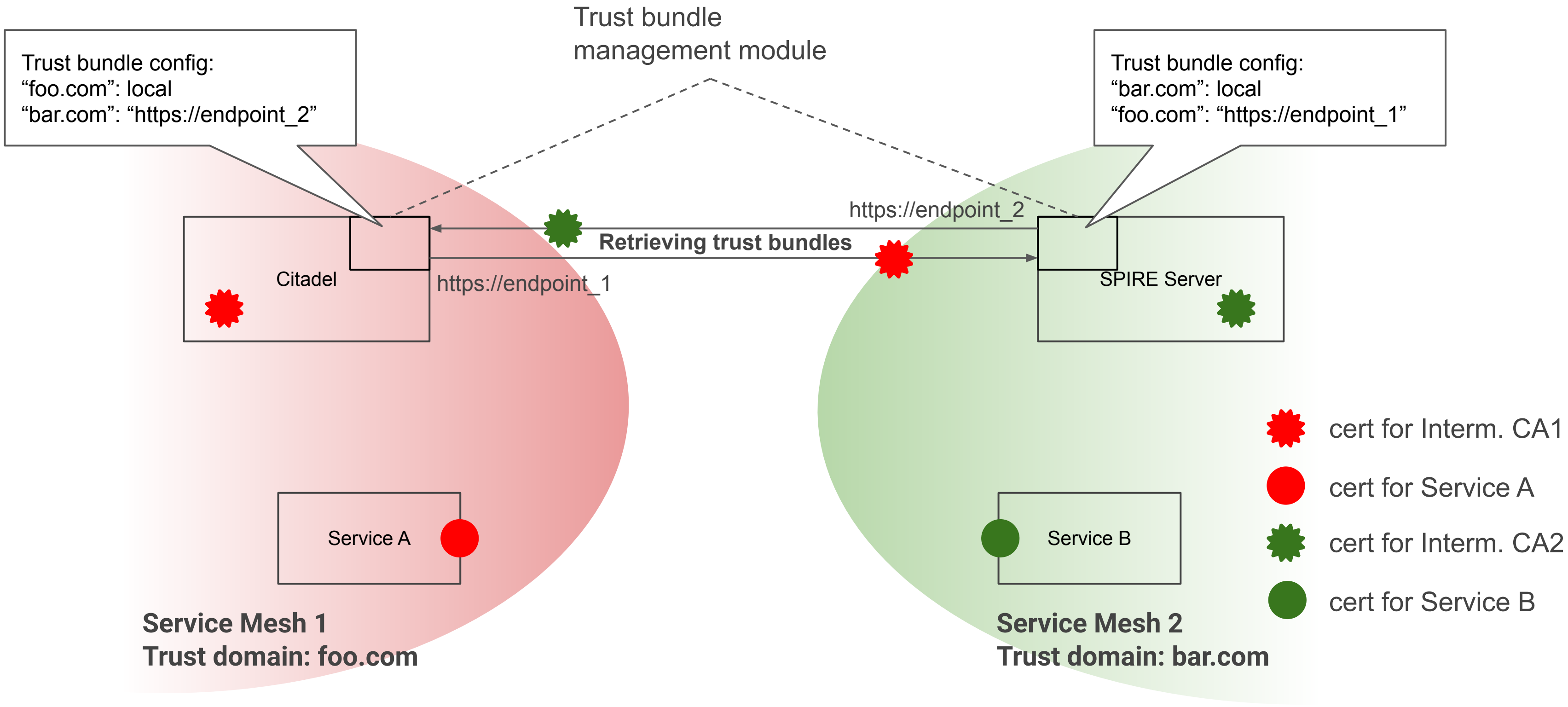
✹ cert for Interm. CA1

● cert for Service A

✹ cert for Interm. CA2

● cert for Service B

Federation with SPIFFE Trust Bundle (2)



Trust bundle management module

Trust bundle config:
“foo.com”: local
“bar.com”: “https://endpoint_2”

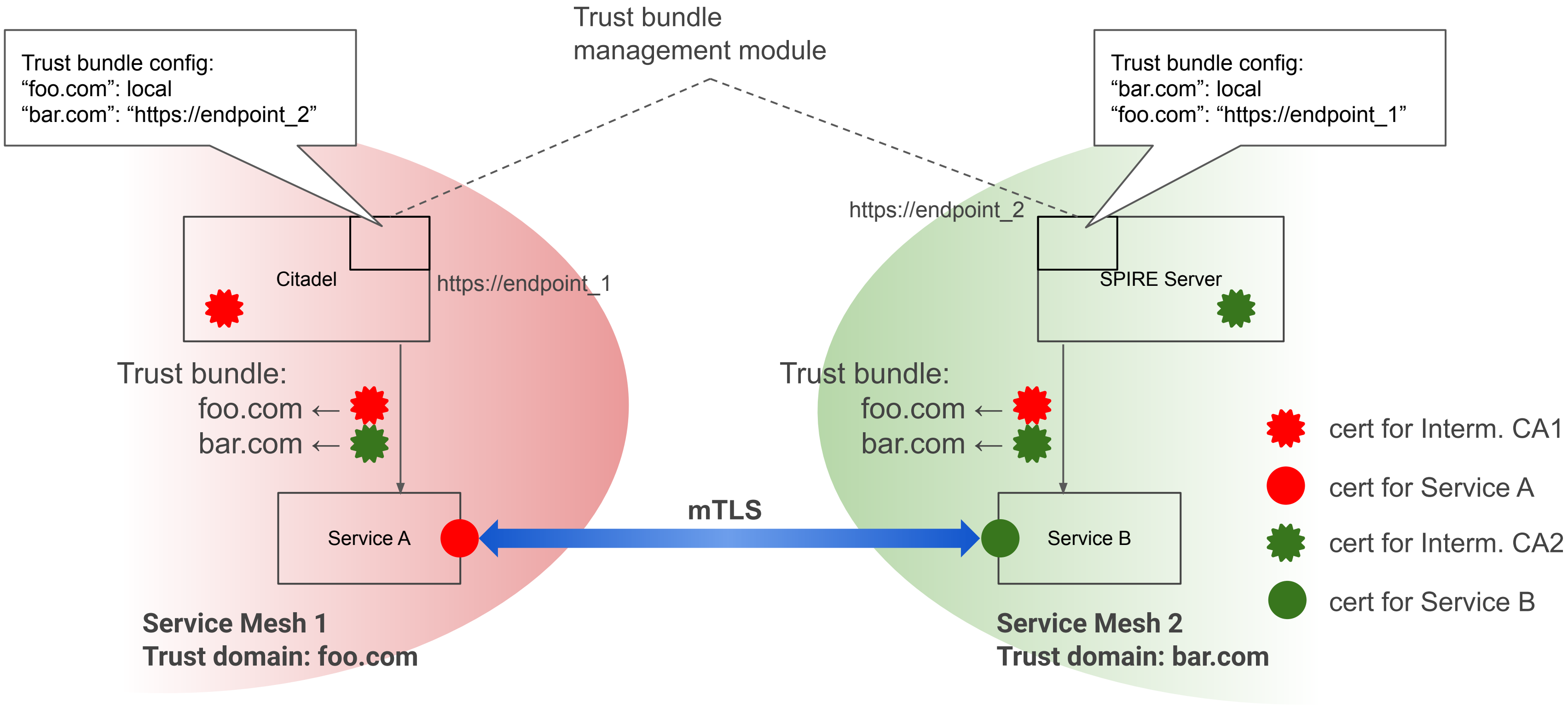
Trust bundle config:
“bar.com”: local
“foo.com”: “https://endpoint_1”

Service Mesh 1
Trust domain: foo.com

Service Mesh 2
Trust domain: bar.com

- ✸ cert for Interm. CA1
- 🔴 cert for Service A
- ✸ cert for Interm. CA2
- 🟢 cert for Service B

Federation with SPIFFE Trust Bundle (2)



Trust bundle management module

Trust bundle config:
“foo.com”: local
“bar.com”: “https://endpoint_2”

Citadel

https://endpoint_1

Trust bundle:
foo.com ←
bar.com ←

Service A

mTLS

Service Mesh 1
Trust domain: foo.com

Trust bundle config:
“bar.com”: local
“foo.com”: “https://endpoint_1”

https://endpoint_2

SPIRE Server

Trust bundle:
foo.com ←
bar.com ←

Service B

Service Mesh 2
Trust domain: bar.com

cert for Interm. CA1

● cert for Service A

cert for Interm. CA2

● cert for Service B

Thank you!
谢谢！

- Istio docs istio.io
- Istio discussion board discuss.istio.io
- Join Istio working groups github.com/istio/community/blob/master/WORKING-GROUPS.md
- Contribute code github.com/istio