Deckhouse
Kubernetes Platform

Istio

Федерация
Общие принципы

cluster-a.local

cluster-b.local

Есть два кластера под управлением Istio...

cluster-a.local

namespace: abc

Pod: foo

Service: foo

cluster-b.local

namespace: xyz

Pod: bar

Service: bar

...В них работают приложения.

cluster-a.local

Certificates

Root CA

namespace: abc

Pod: foo

Service: foo

cluster-b.local

Certificates

Root CA

namespace: xyz

Pod: bar

Service: bar

У каждого кластера есть хранилище доверенных сертификатов, которое содержит единственный корневой сертификат кластера.

cluster-a.local

Certificates

Root CA

namespace: abc

Pod: foo

Service: foo

cluster-b.local

Certificates

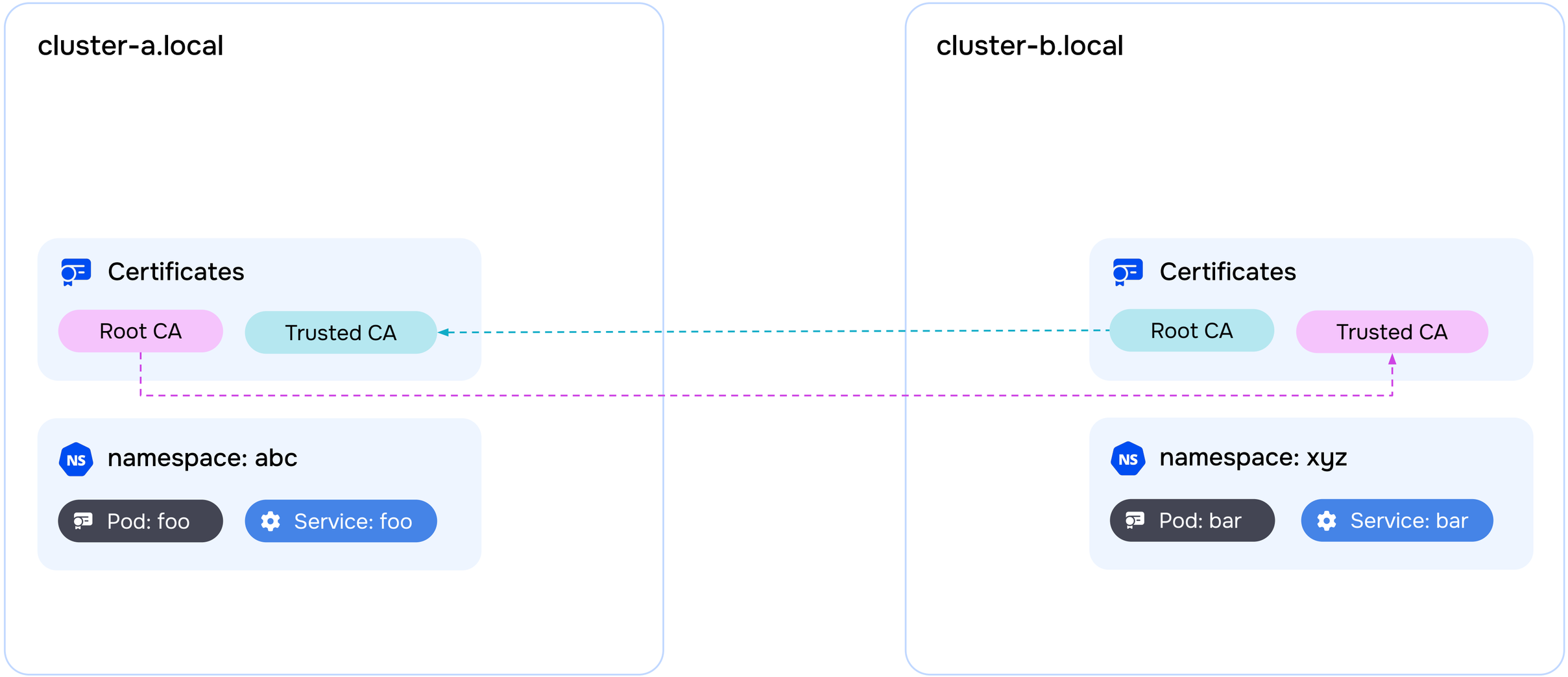
Root CA

namespace: xyz

Pod: bar

Service: bar

Этими корневыми сертификатами подписаны индивидуальные сертификаты подов для нужд Mutual TLS.



Для обеспечения взаимного доверия необходимо взаимно обменяться корневыми сертификатами и поместить их в хранилище доверенных сертификатов.

cluster-a.local

Certificates

Root CA

Trusted CA

namespace: abc

Pod: foo

Service: foo

Cloud

Ingress-gateway

Ingress-gateway

cluster-b.local

Certificates

Root CA

Trusted CA

namespace: xyz

Pod: bar

Service: bar

У каждого кластера есть ingress-gateway, который позволяет принимать Mutual TLS запросы извне кластера.

cluster-a.local

Certificates

Root CA

Trusted CA

namespace: abc

Pod: foo

Service: foo

Cloud

Ingress-gateway

Ingress-gateway

cluster-b.local

Certificates

Root CA

Trusted CA

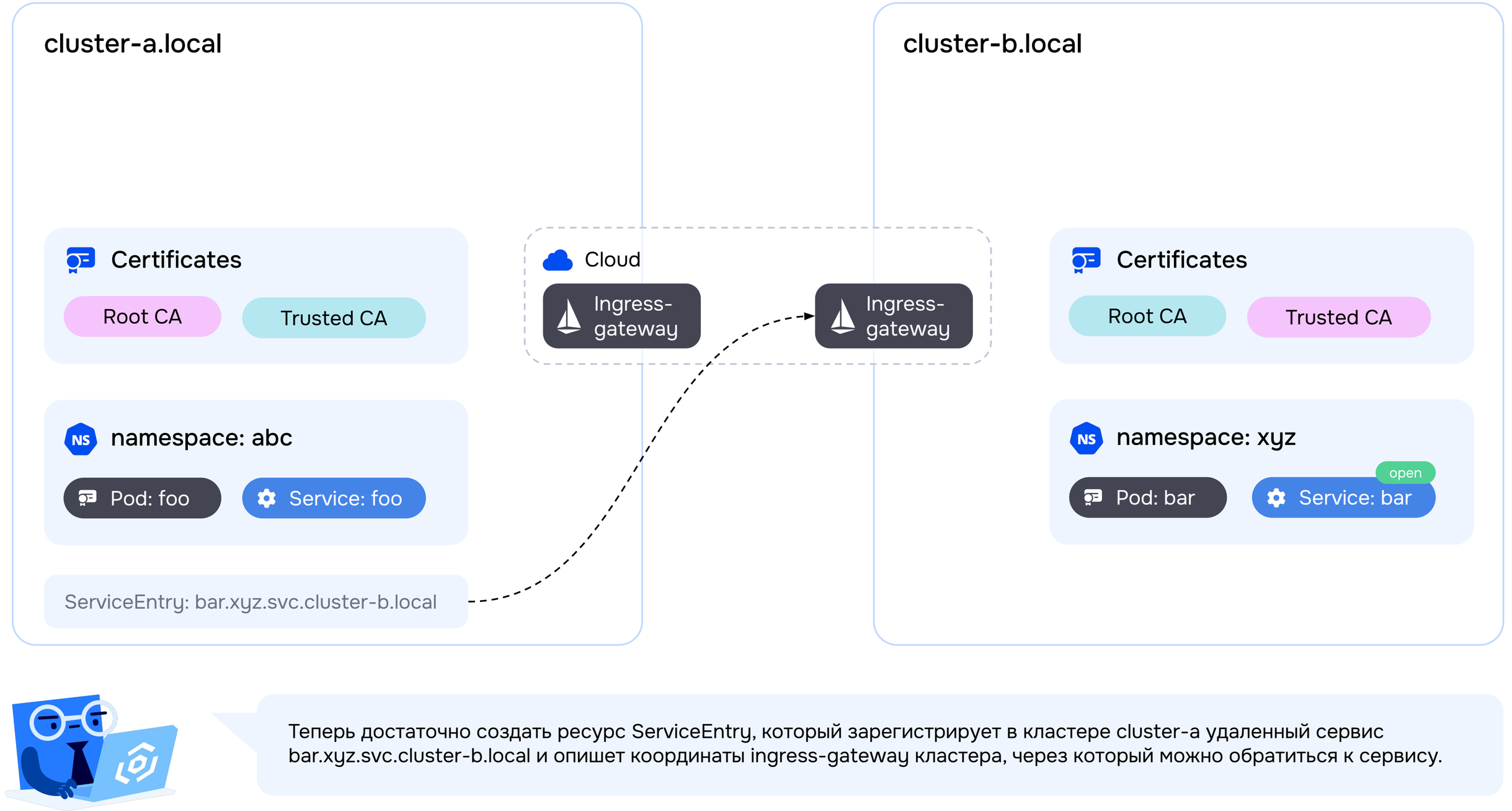
namespace: xyz

Pod: bar

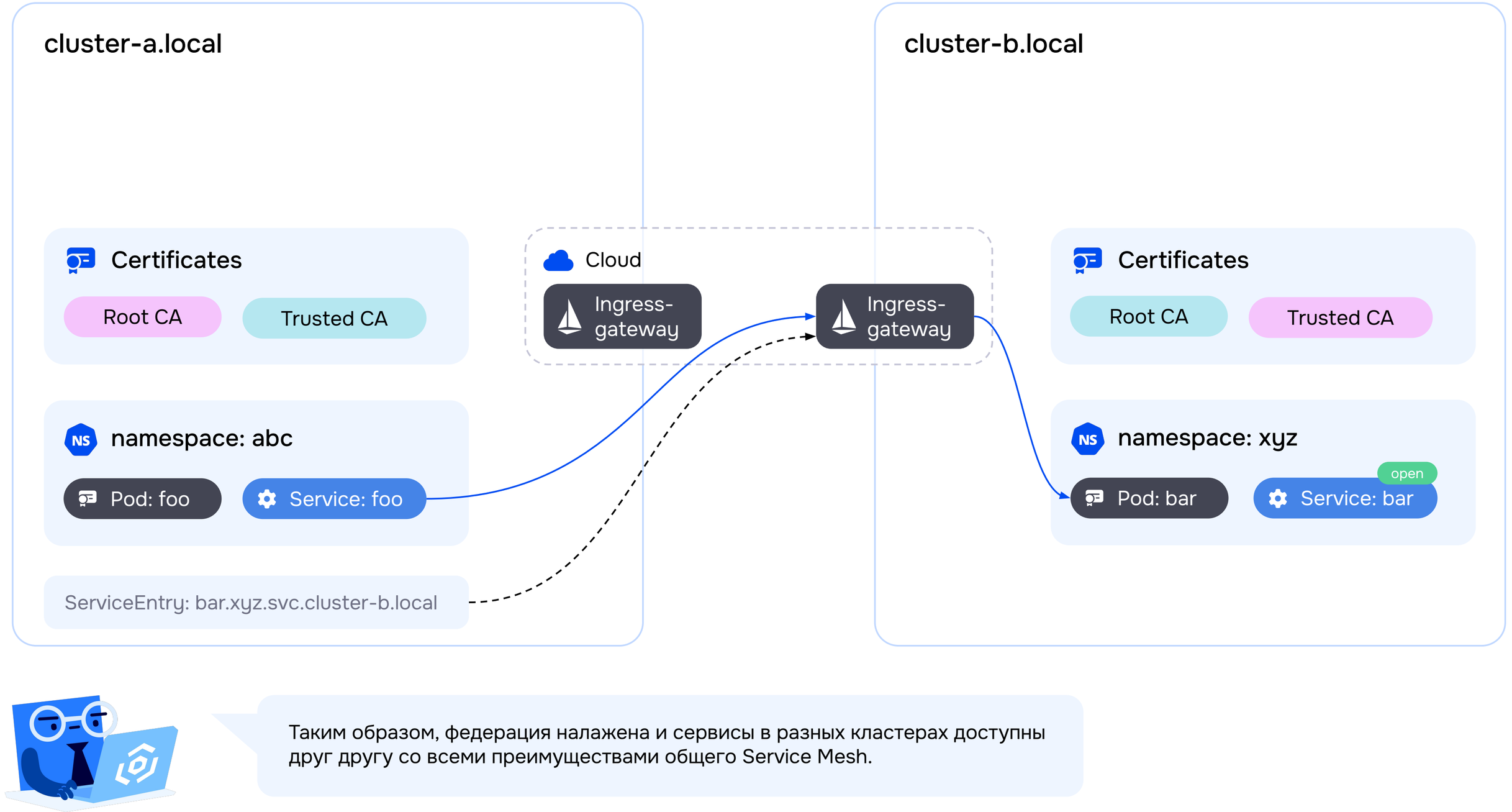
open

Service: bar

Мы используем эти ingress-gateway, чтобы предоставить доступ к сервису внешним кластерам в рамках федерации.



Теперь достаточно создать ресурс ServiceEntry, который зарегистрирует в кластере cluster-a удаленный сервис bar.xyz.svc.cluster-b.local и опишет координаты ingress-gateway кластера, через который можно обратиться к сервису.



Таким образом, федерация налажена и сервисы в разных кластерах доступны друг другу со всеми преимуществами общего Service Mesh.