

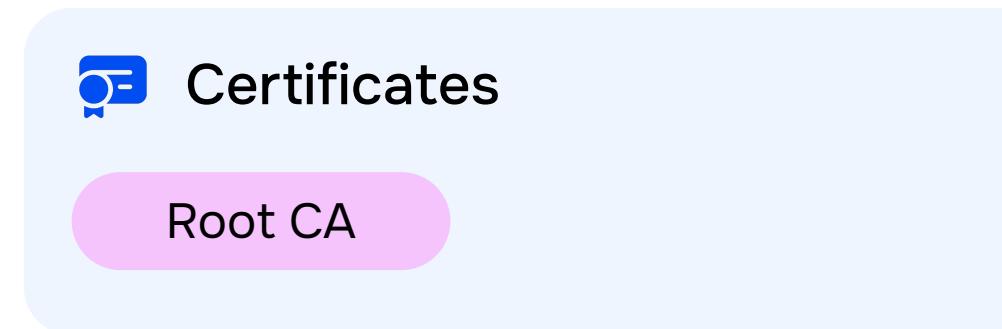


Deckhouse
Kubernetes Platform

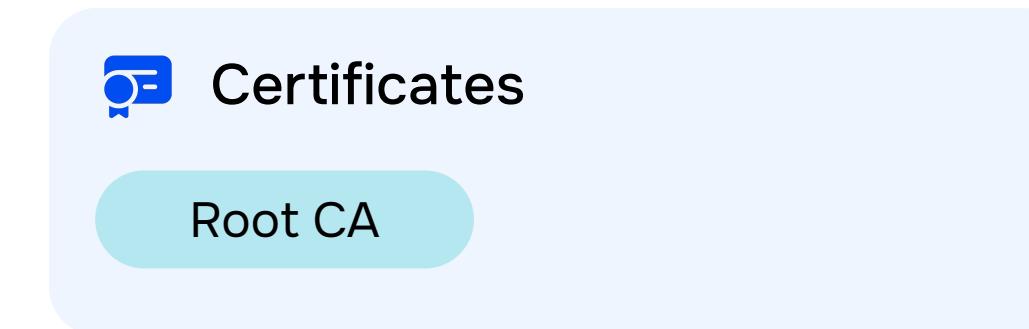
Istio

Мультиклuster
IstioMulticloud

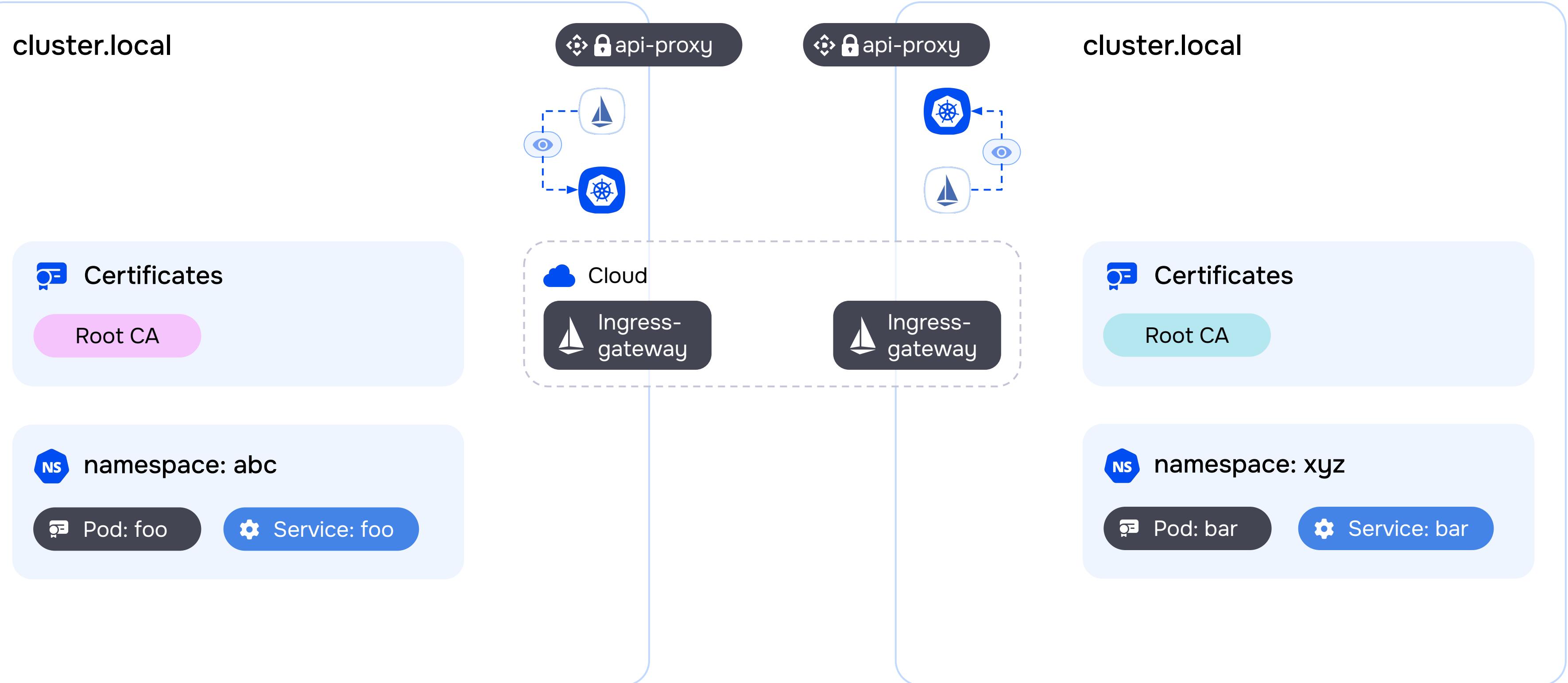
cluster.local



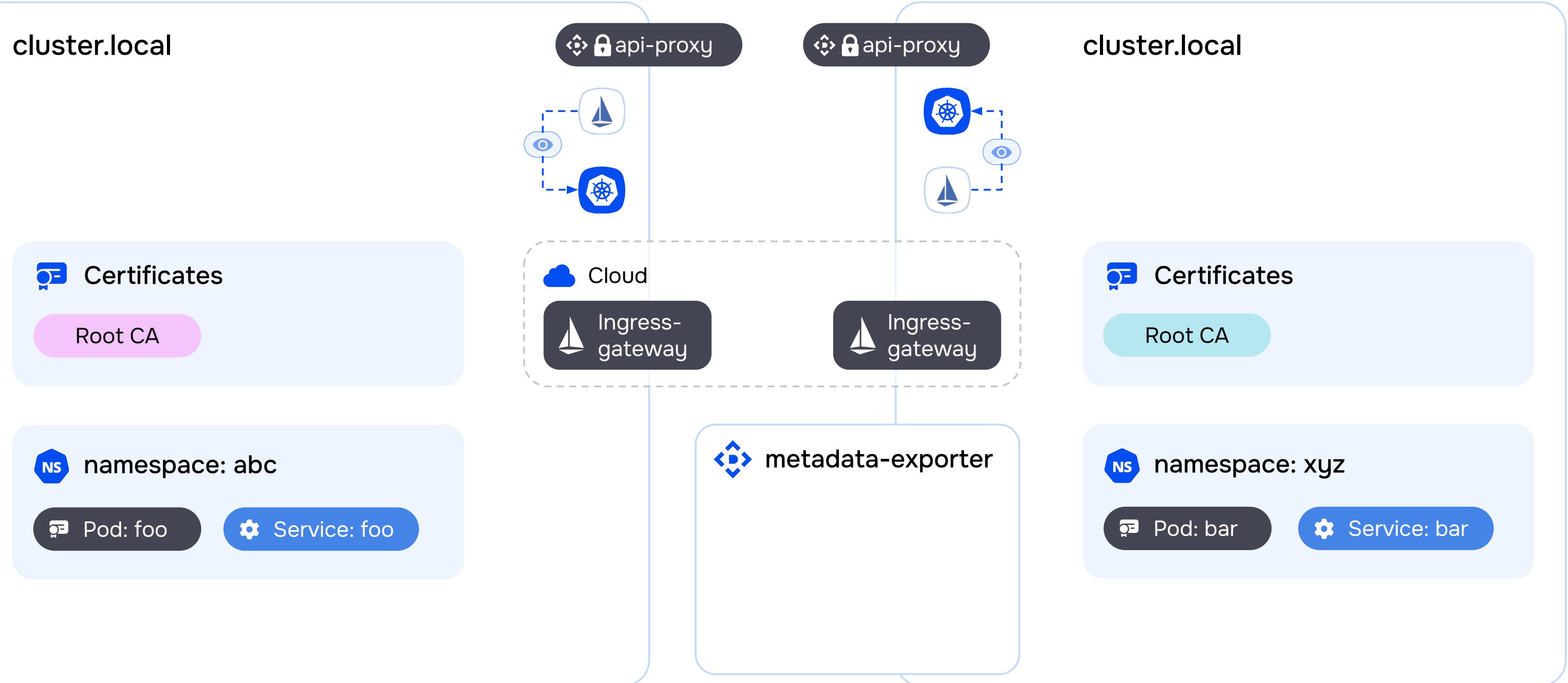
cluster.local



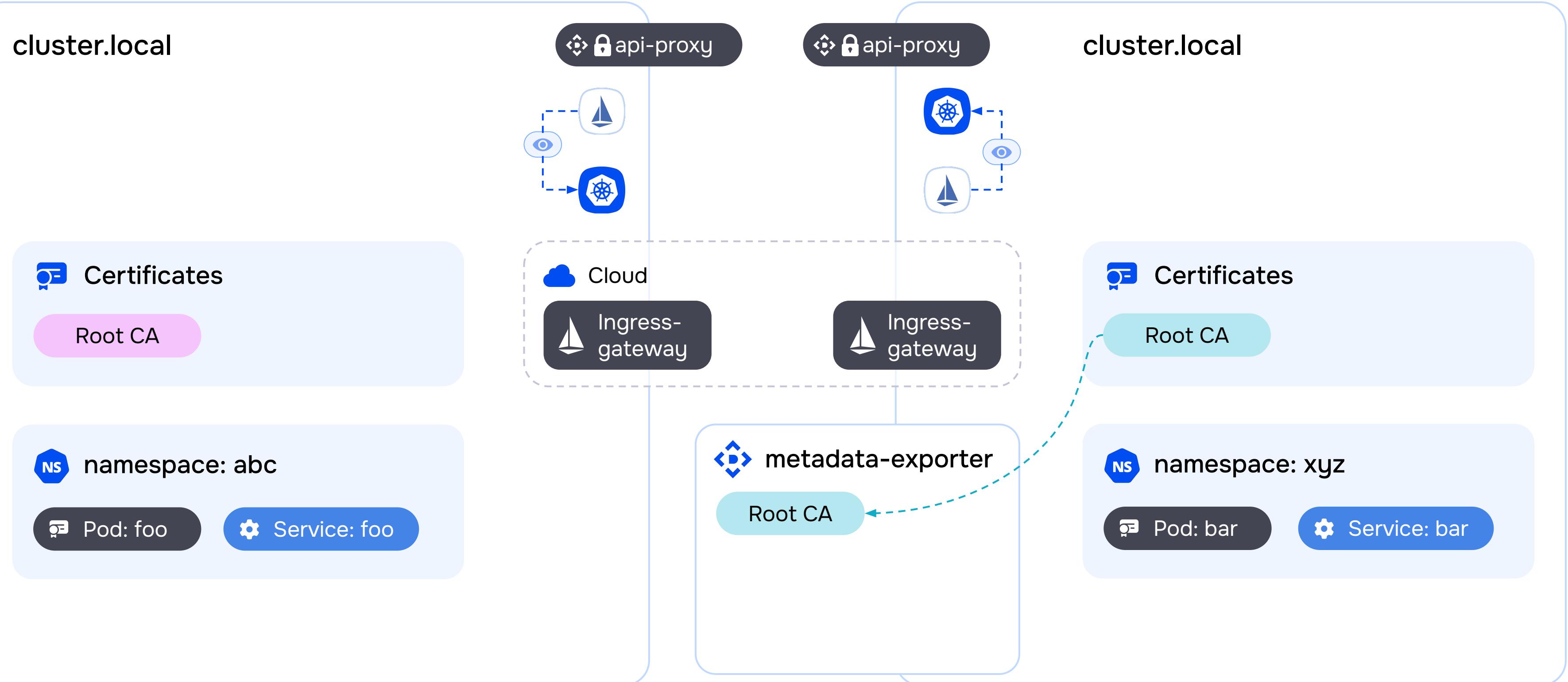
Есть два кластера под управлением Istio с одинаковым Cluster Domain. У каждого кластера свой корневой istio-сертификат. Требуется объединить два кластера в единый Service Mesh.



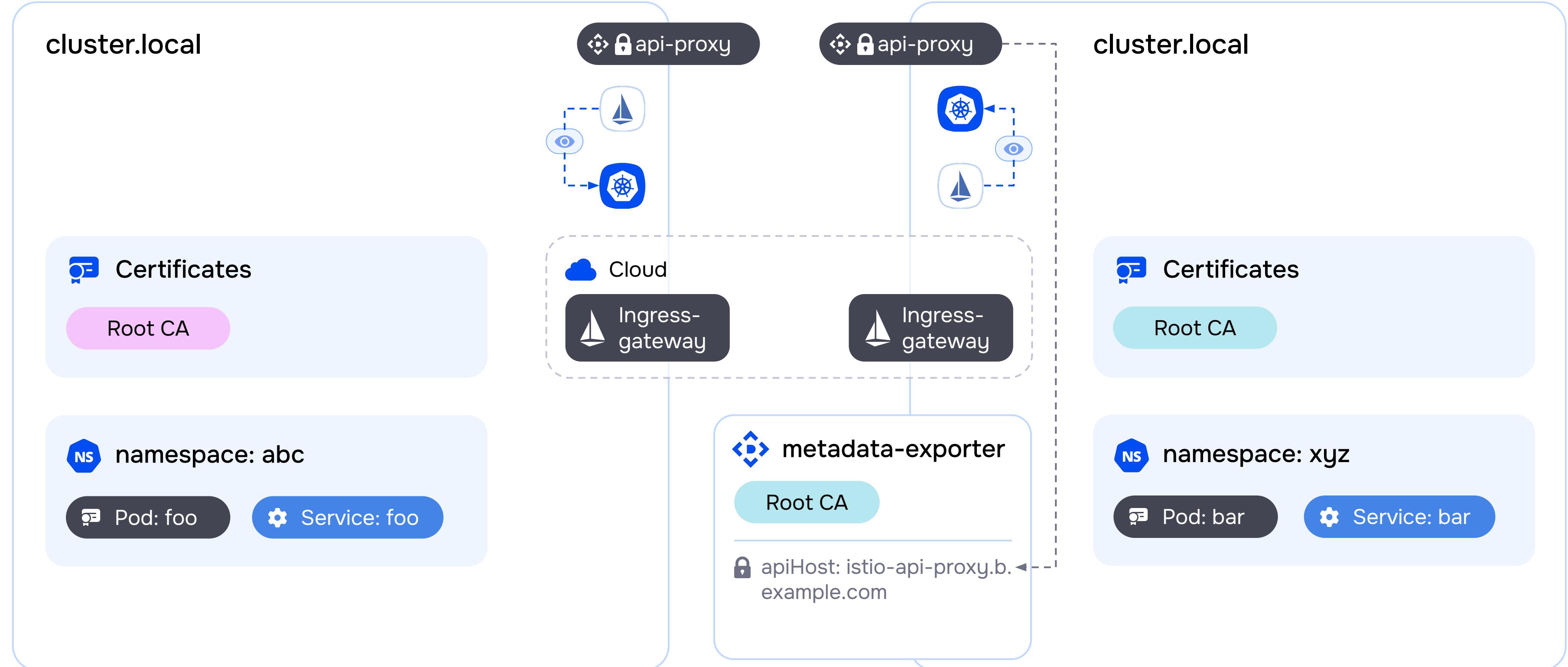
Включаем параметр модуля `istio.multicloud.enabled = true`, после чего появляется компонент `api-proxy` для предоставления удаленным кластерам ограниченного доступа к `apiserver`...



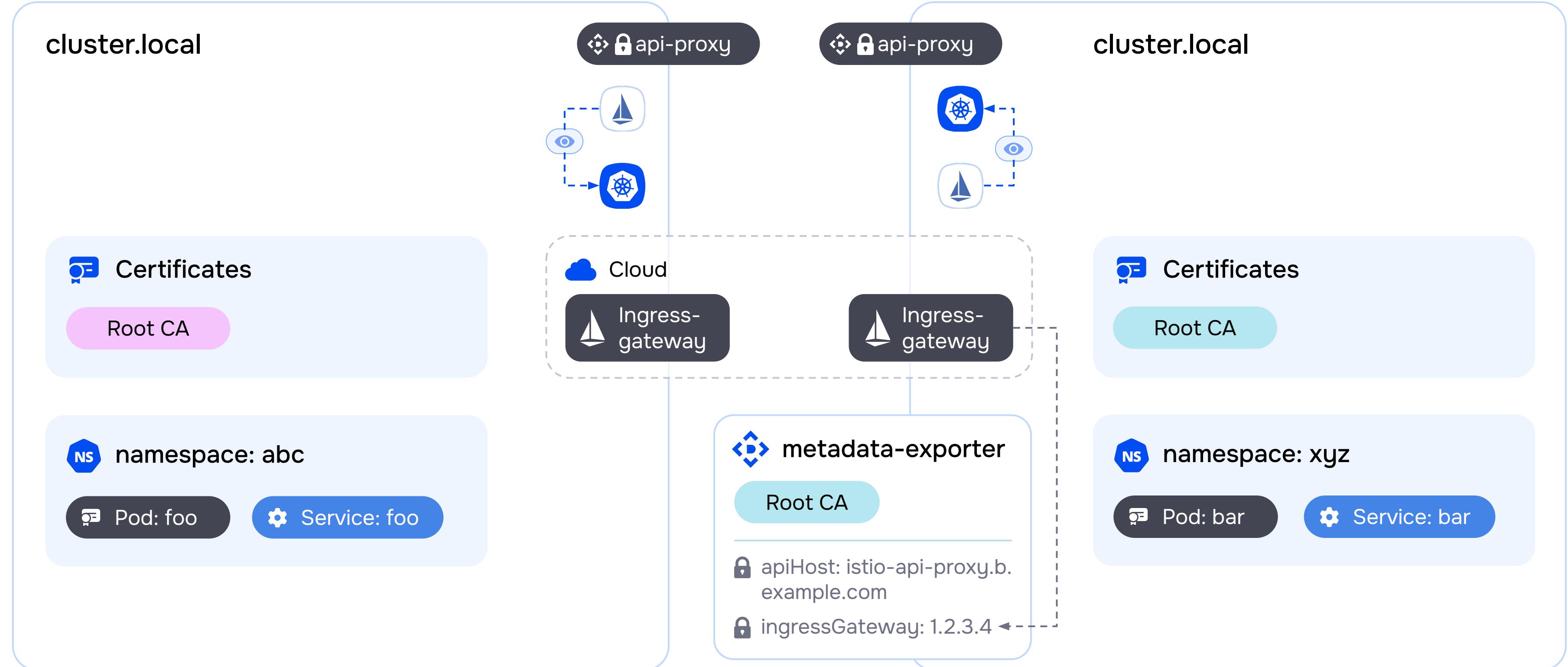
...также, запускается компонент **metadata-exporter**, который собирает и публикует метаинформацию о кластере...
(для примера проиллюстрируем только на правом кластере, действие происходит на обоих кластерах)



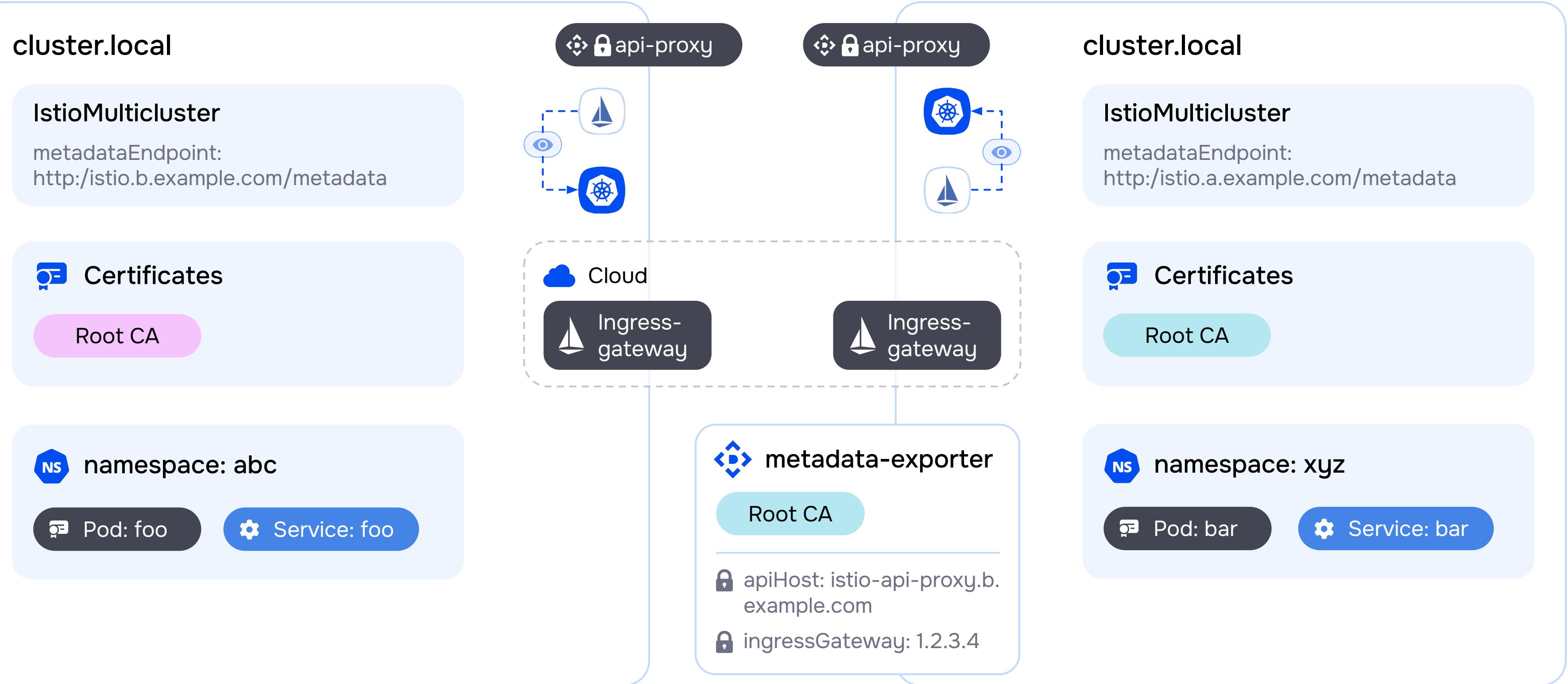
публичную часть корневого istio-сертификата...



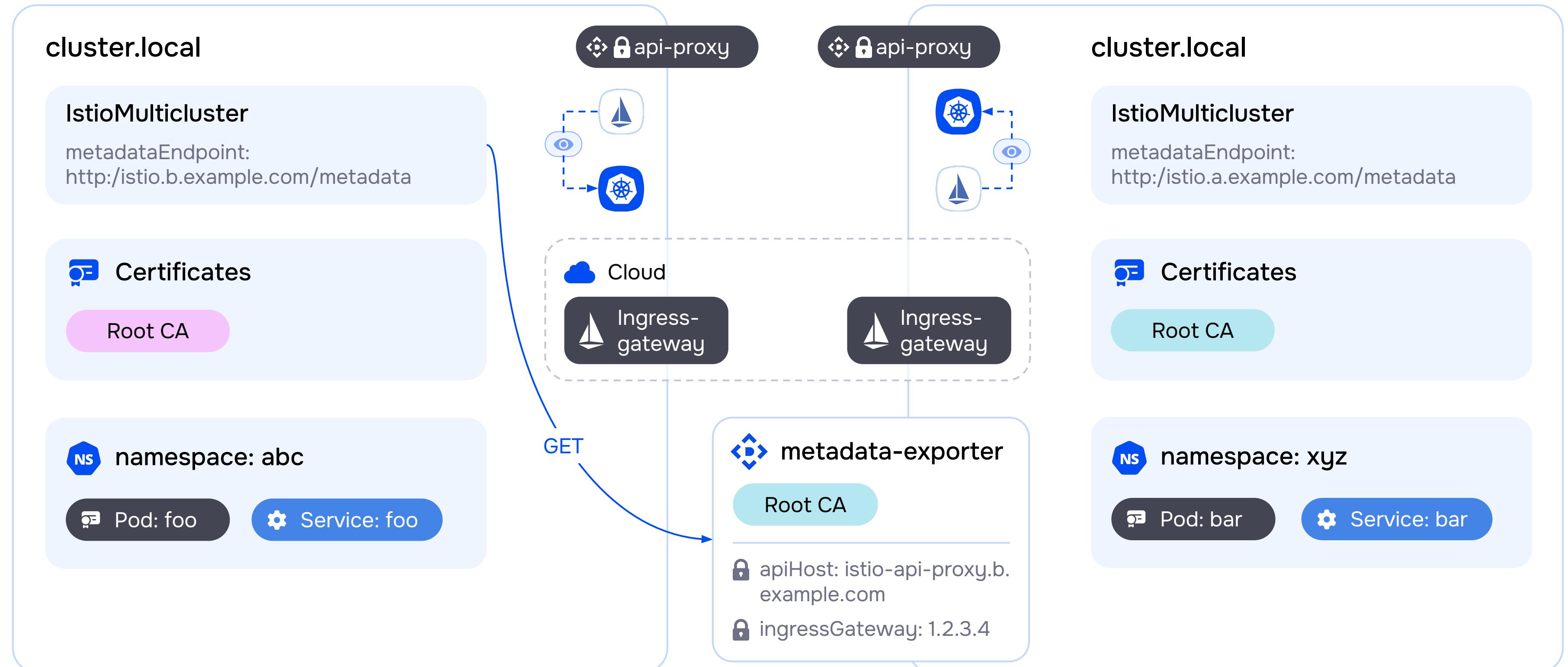
адрес прокси для доступа в apiserver...



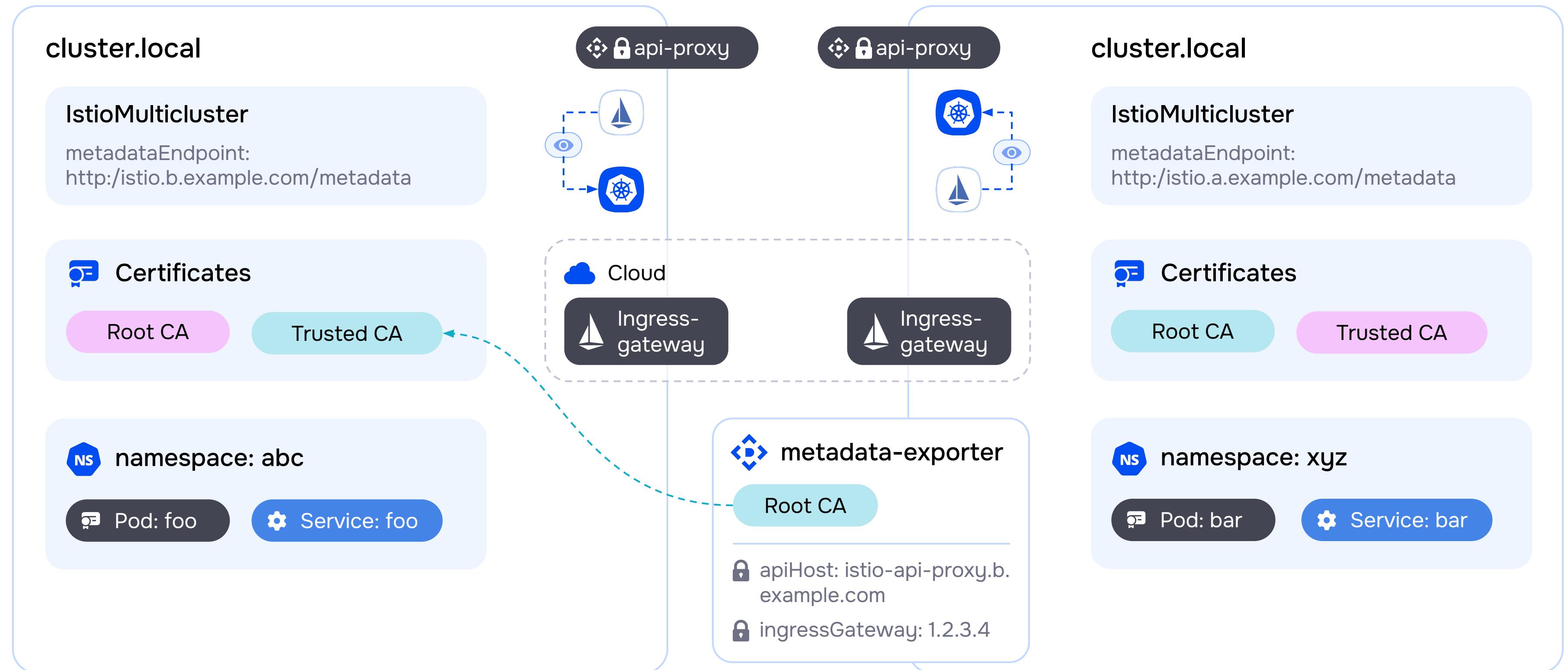
публичные адреса компонентов ingress-gateway и прочую метаинформацию.



На кластерах взаимно создаются ресурсы IstioMulticloud, которые обозначают координаты с метаданными удаленного кластера, далее организация мультиклестера происходит автоматически...



Deckhouse собирает удаленные метаданные...



скачивает публичный корневой сертификат и обменивается ключами для доступа к закрытым метаданным...

cluster.local

IstioMulticloud

metadataEndpoint:
<http://istio.b.example.com/metadata>

Certificates

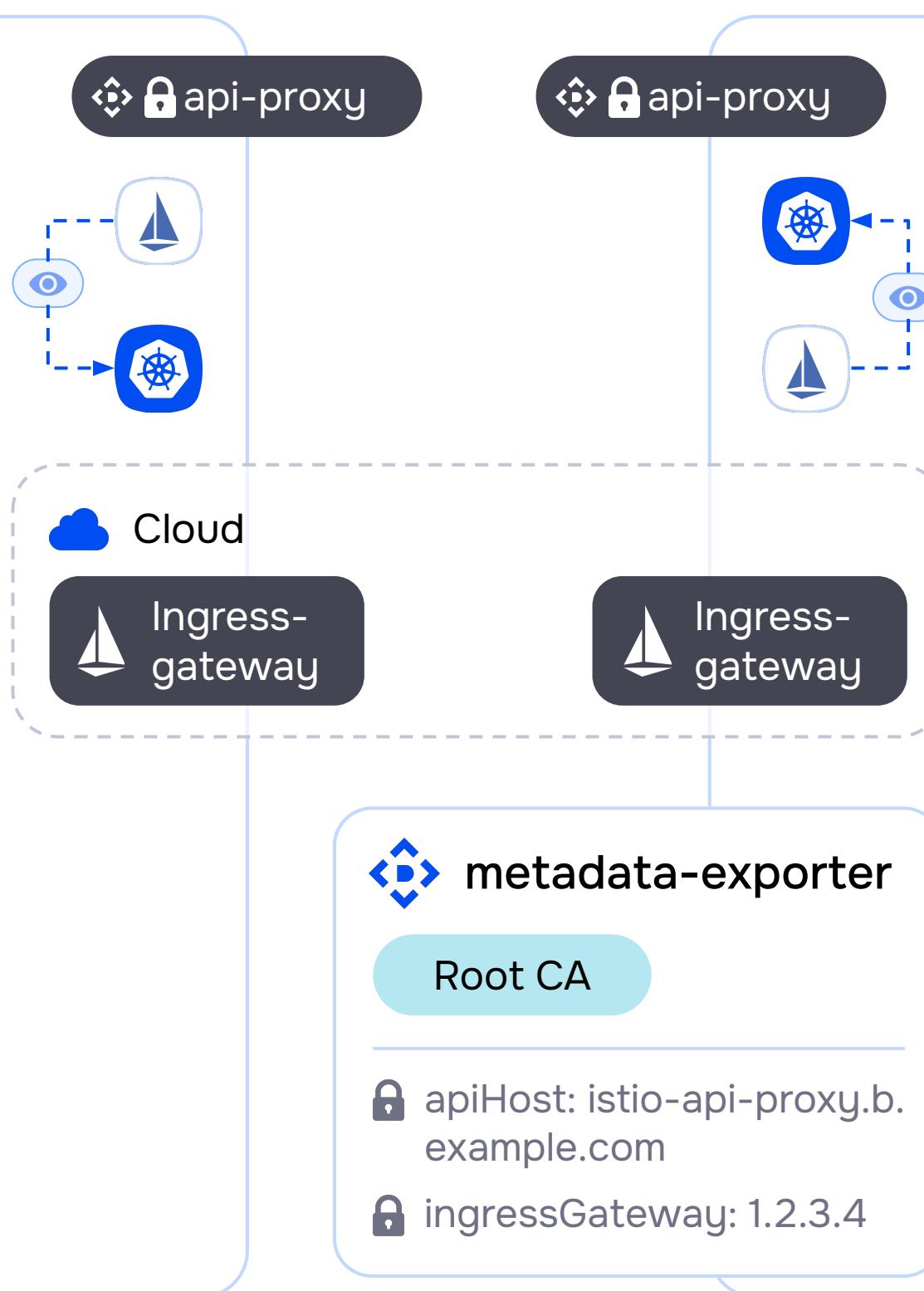
Root CA

Trusted CA

namespace: abc

Pod: foo

Service: foo



cluster.local

IstioMulticloud

metadataEndpoint:
<http://istio.a.example.com/metadata>

Certificates

Root CA

Trusted CA



скачивает информацию об адресах api-proxy для доступа к удаленному apiserver и адресах ingress-gateway, через которые доступны приложения на удаленном кластере...

cluster.local

IstioMulticloud

metadataEndpoint:
<http://istio.b.example.com/metadata>

Certificates

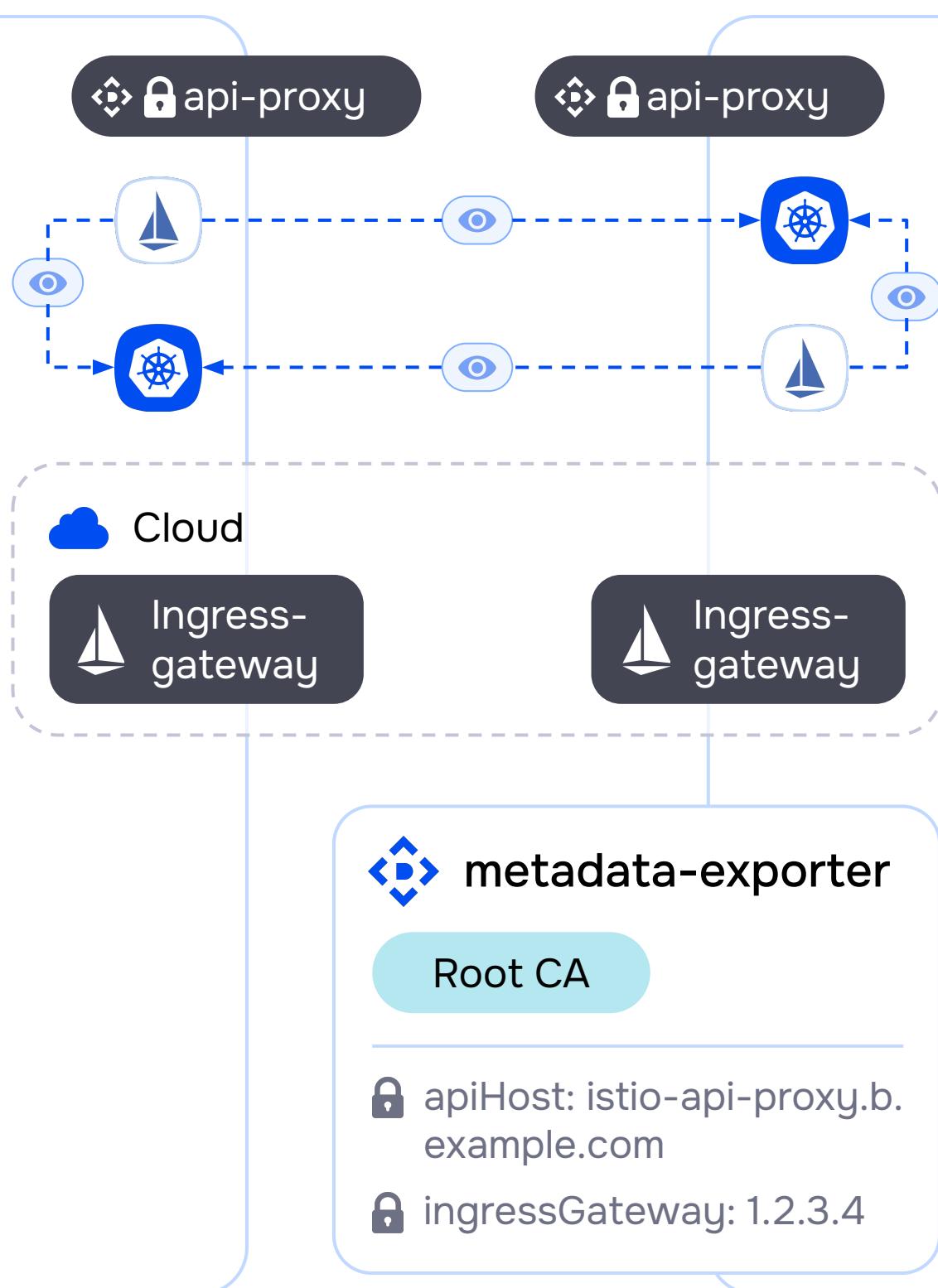
Root CA

Trusted CA

namespace: abc

Pod: foo

Service: foo



cluster.local

IstioMulticloud

metadataEndpoint:
<http://istio.a.example.com/metadata>

Certificates

Root CA

Trusted CA

namespace: xyz

Pod: bar

Service: bar



после чего control plane Istio устанавливает связь с удаленным apiserver...

cluster.local

IstioMulticloud

metadataEndpoint:
<http://istio.b.example.com/metadata>

Certificates

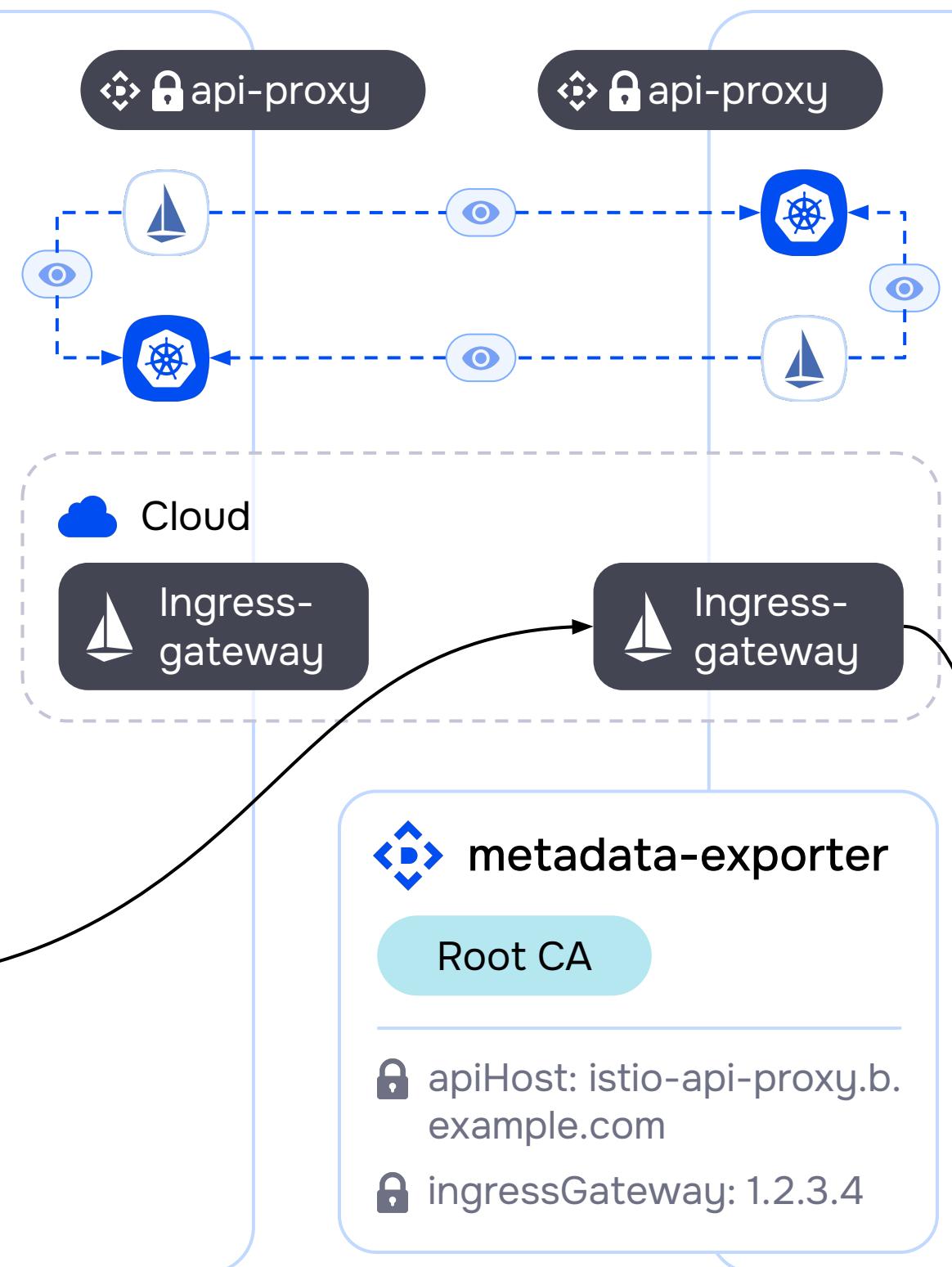
Root CA

Trusted CA

namespace: abc

Pod: foo

Service: foo



cluster.local

IstioMulticloud

metadataEndpoint:
<http://istio.a.example.com/metadata>

Certificates

Root CA

Trusted CA



и приложения в кластерах получают взаимный доступ