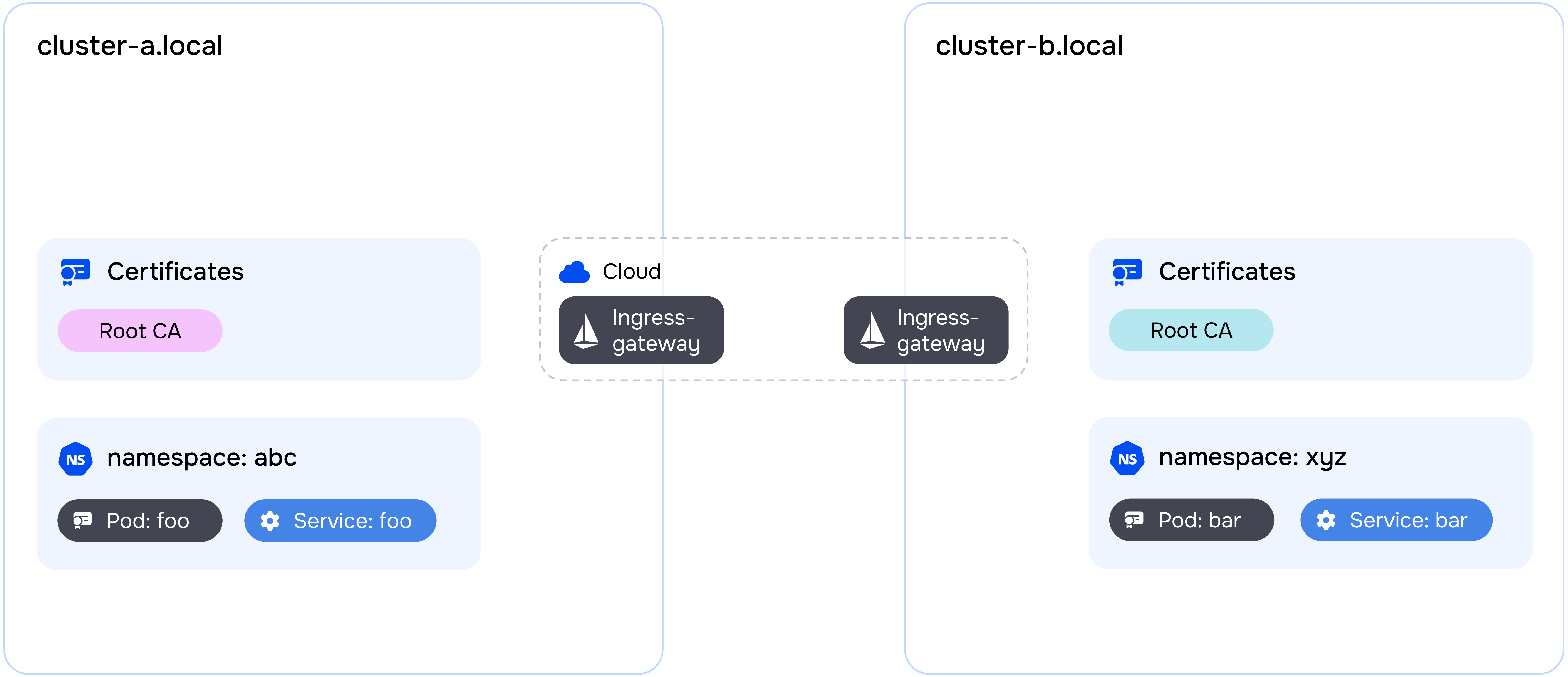
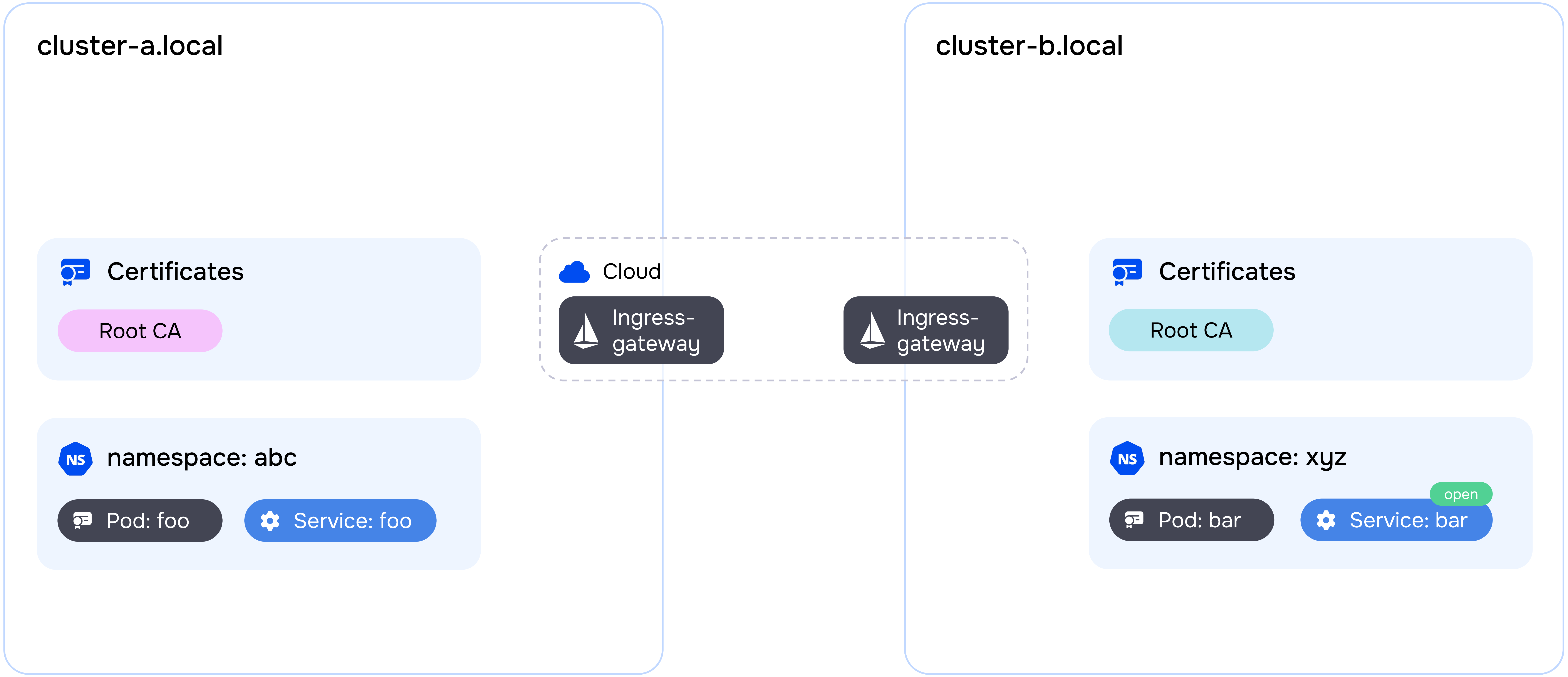
Deckhouse
Kubernetes Platform

# Istio

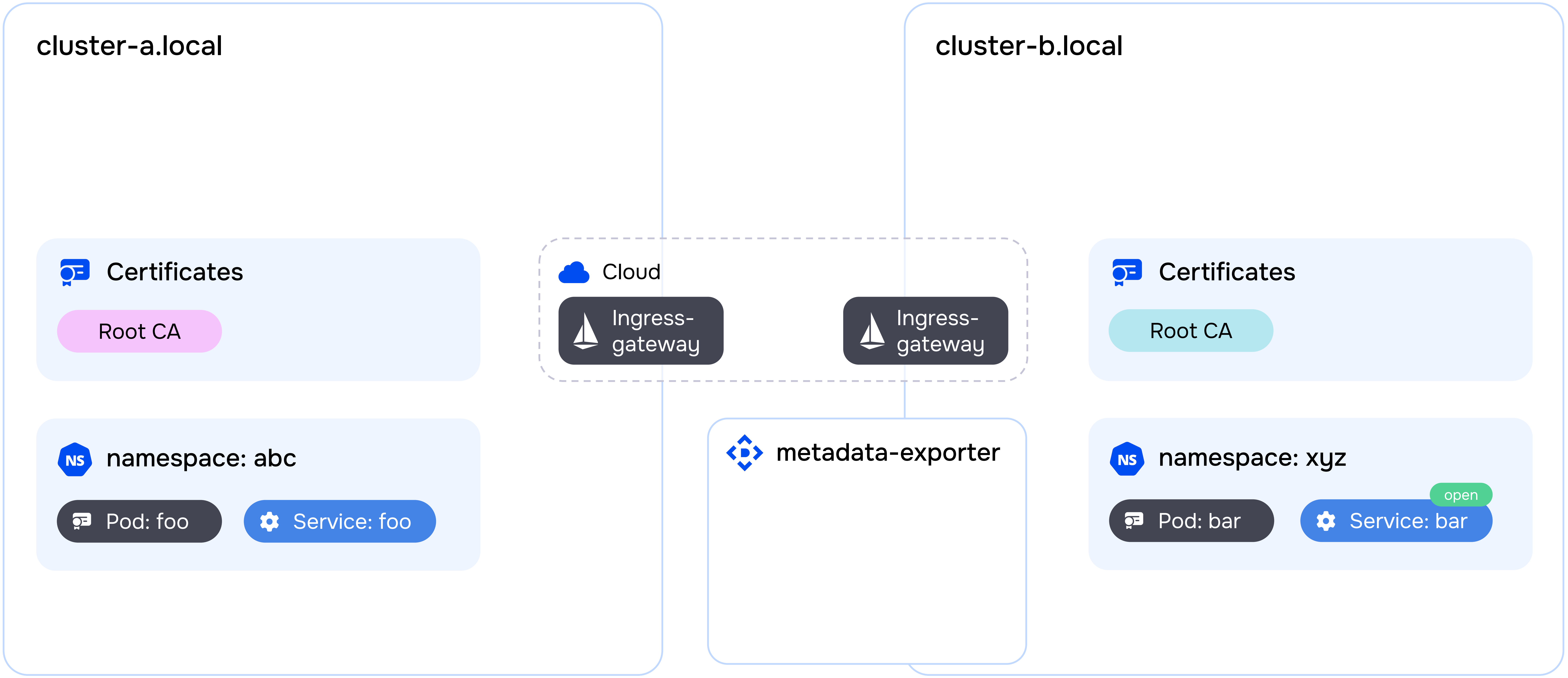
Federation
IstioFederation



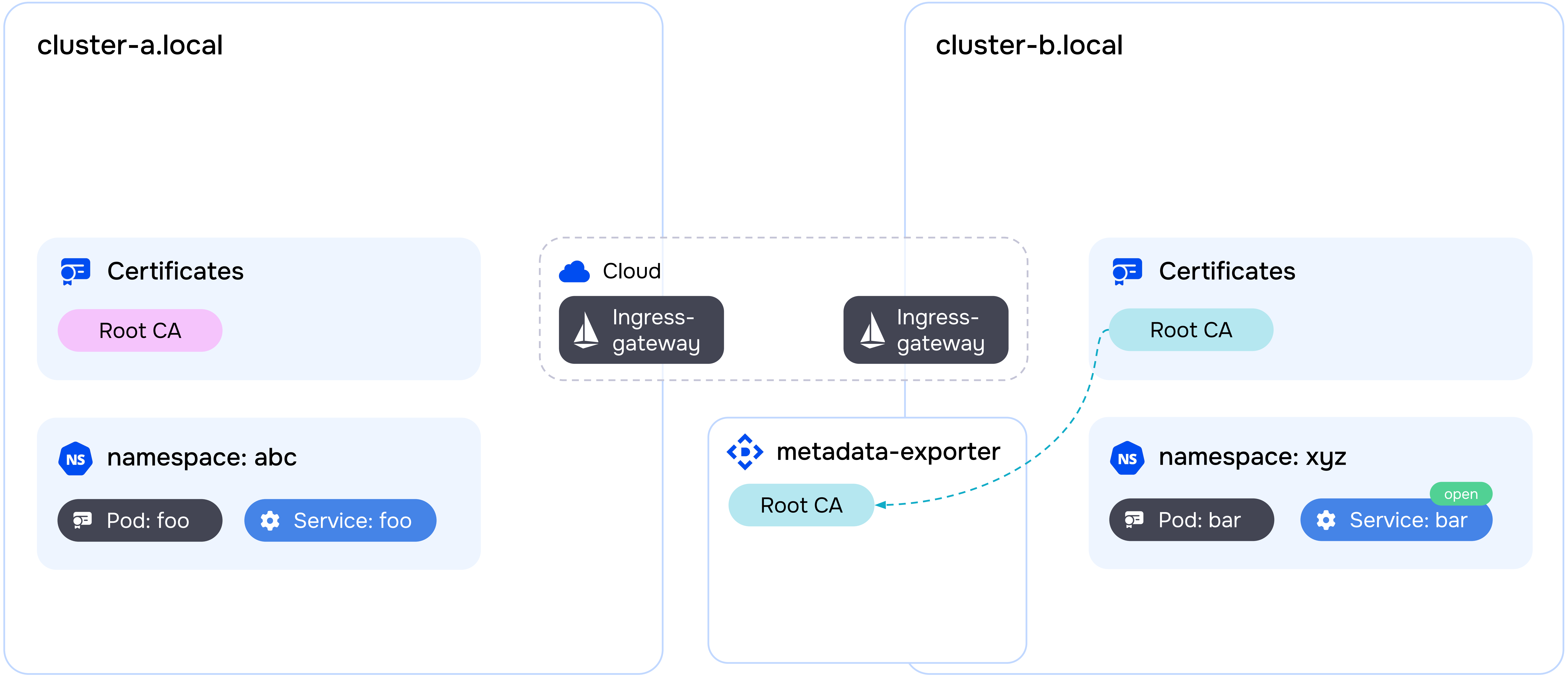
Suppose there are two independent clusters with applications running in them. Each cluster has its own Istio root certificate for signing individual Pod certificates for Mutual TLS needs.



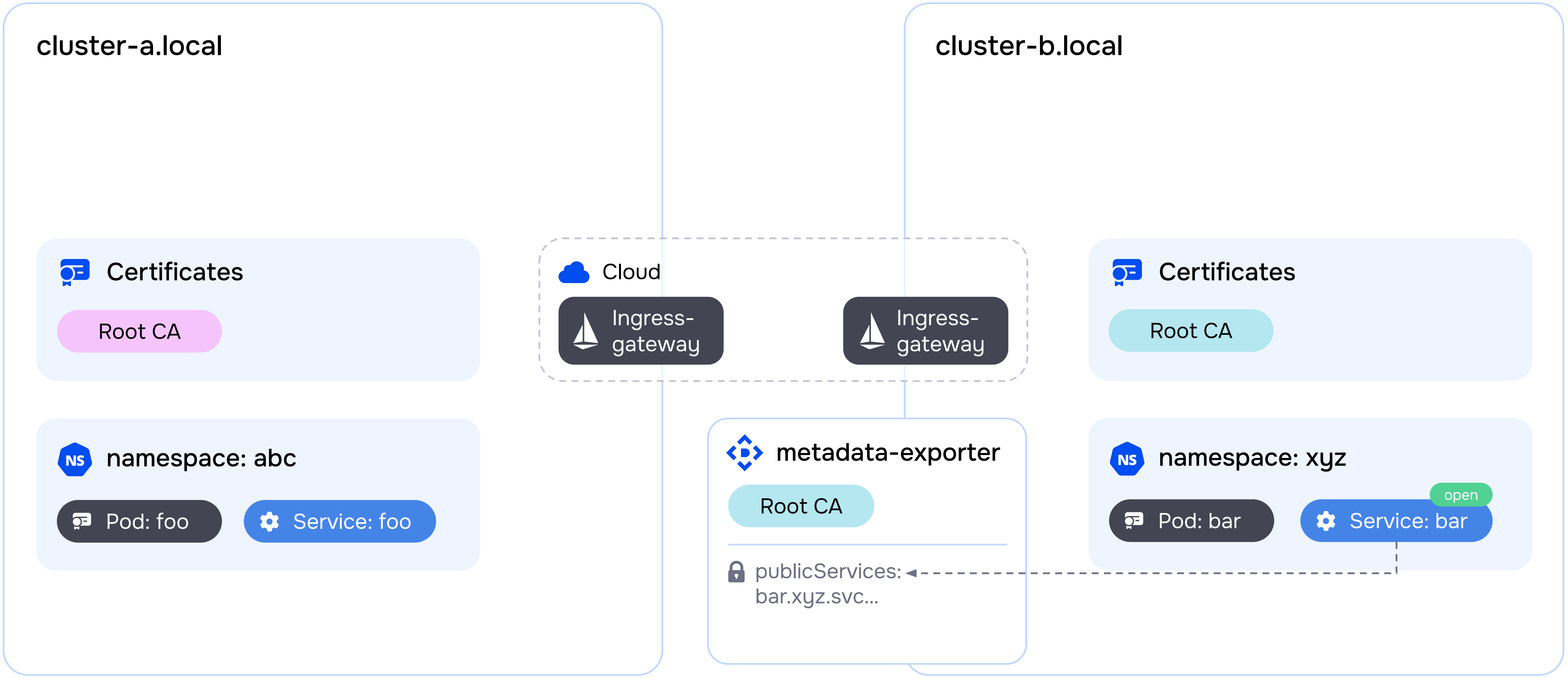
Now, you need to set up a federation and share the bar.xyz.svc.cluster-b.local service between the clusters. To do this, set the istio.federation.enabled of the istio.federation module to true (istio.federation.enabled = true).



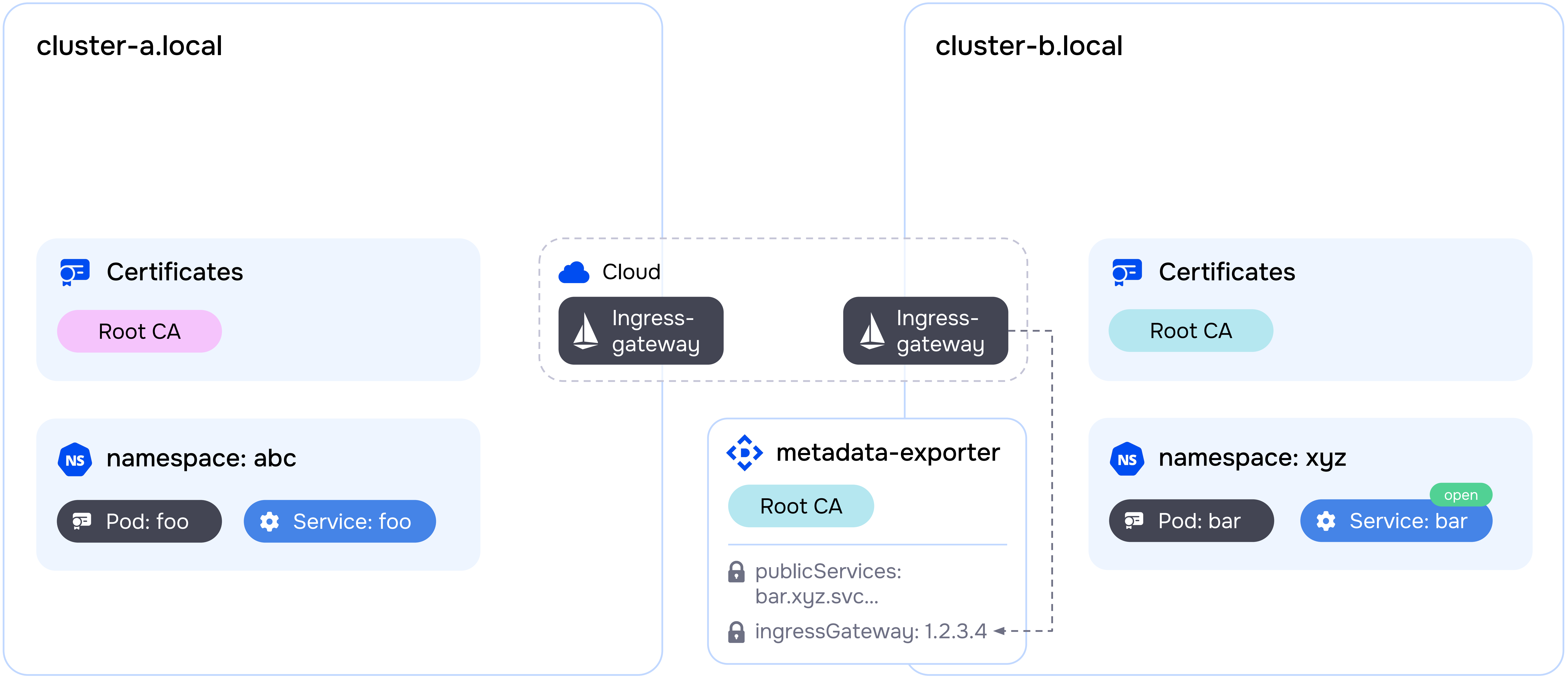
The metadata-exporter component collects and publishes meta-information such as: (Note that we will illustrate the process for cluster-b.local only. However, it is the same for both clusters.)



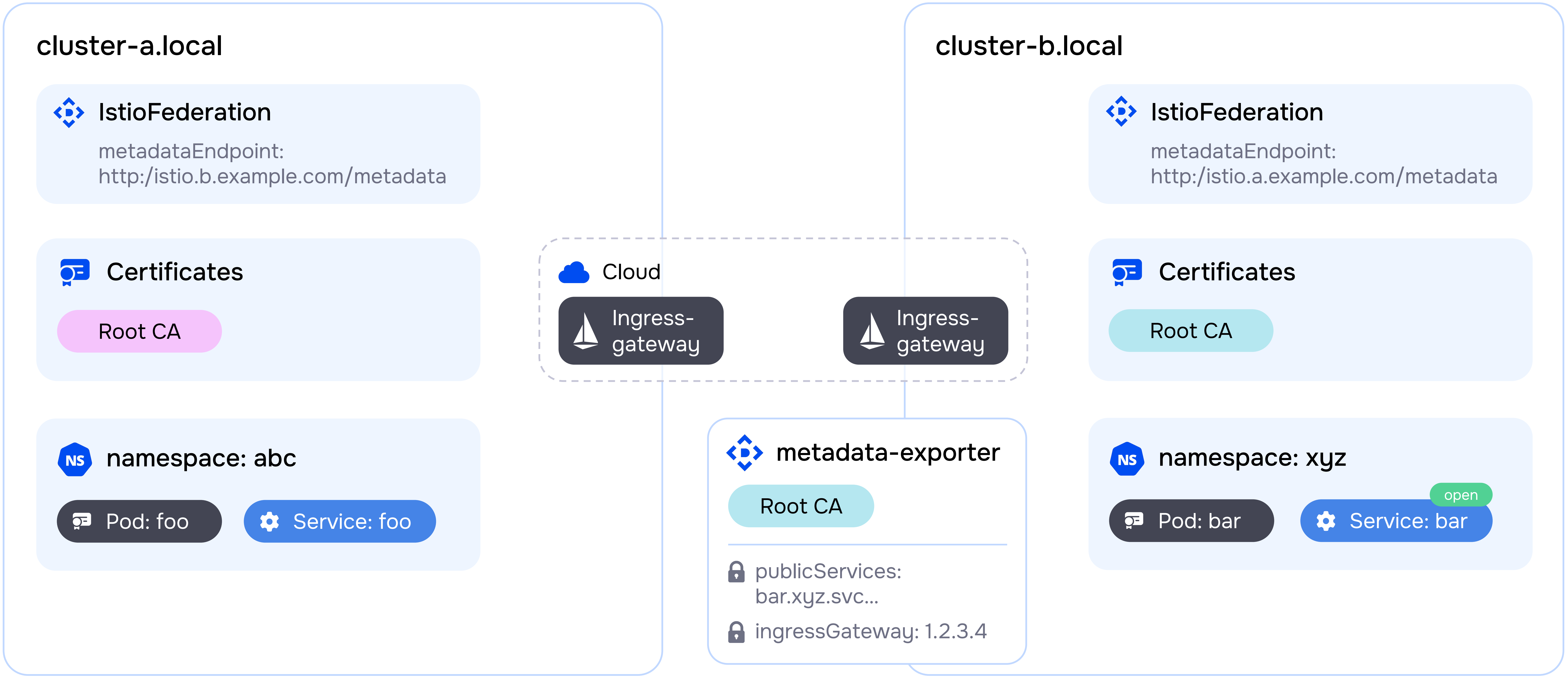
- the public part of the Istio root certificate;



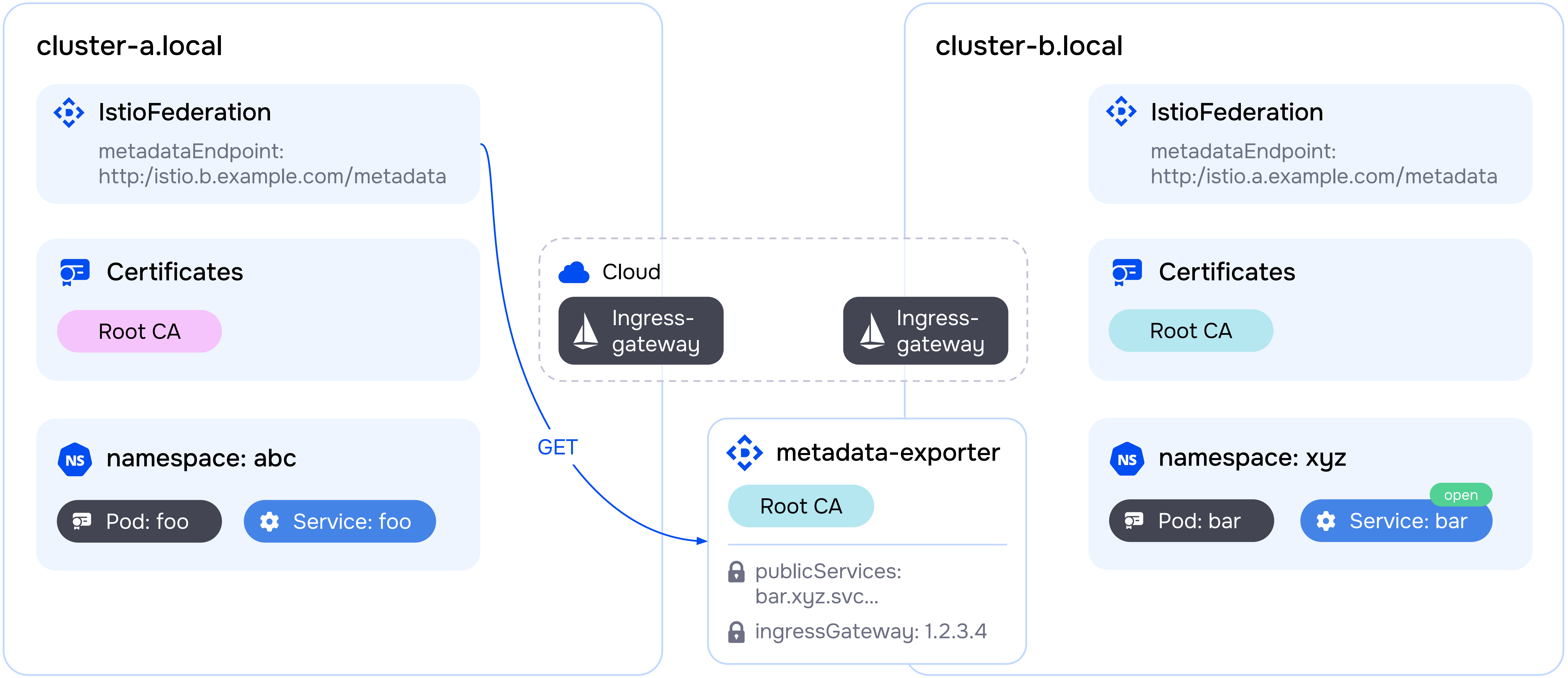
- the list of public services (are only accessible from the federated clusters);



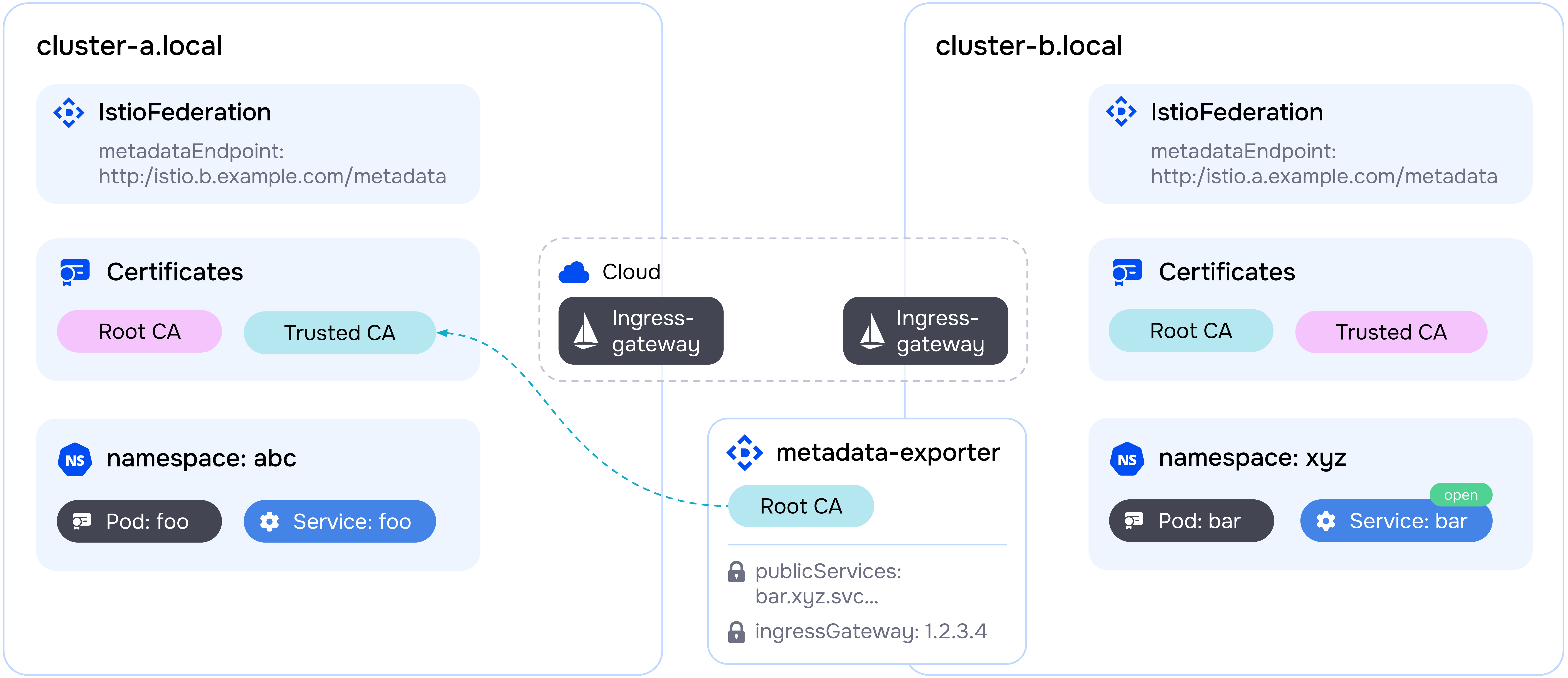
- public addresses of ingress gateway components (are only accessible from the federated clusters)



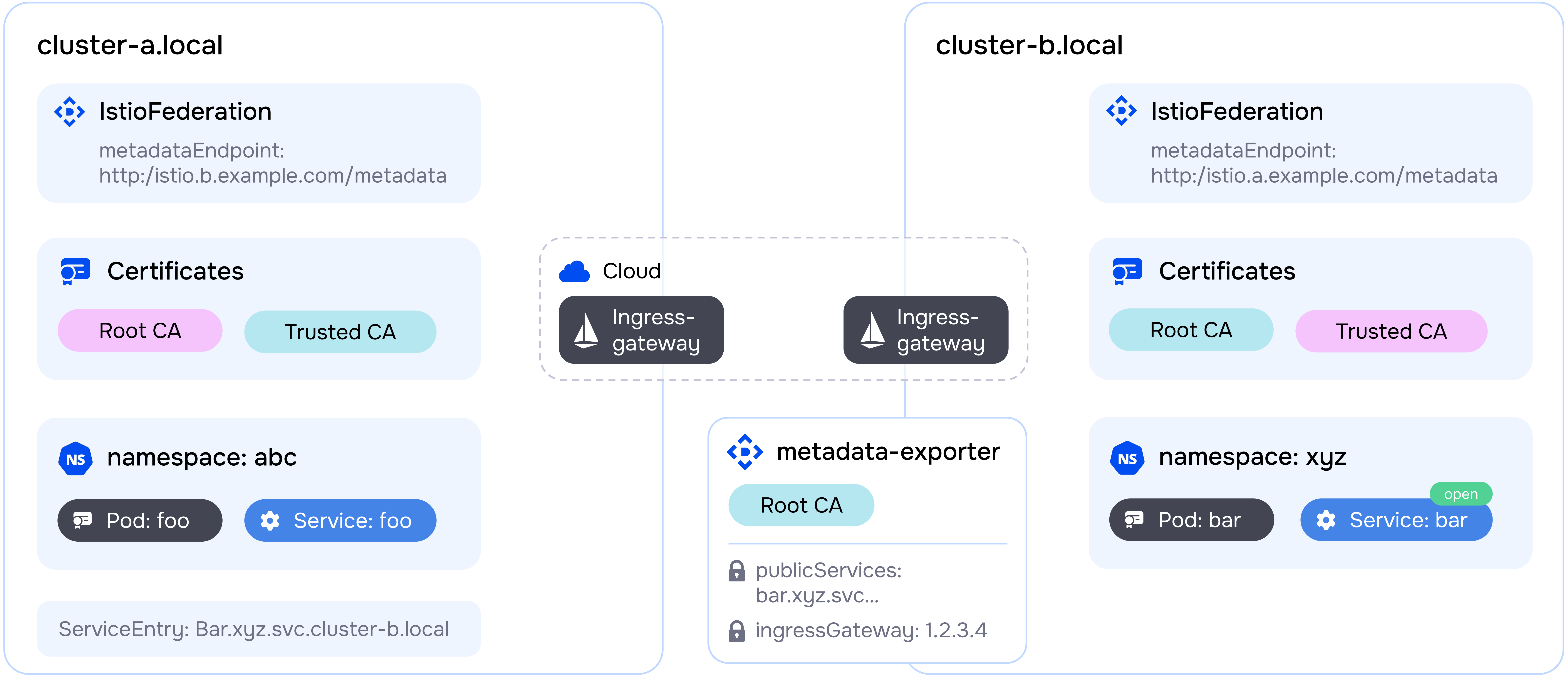
The IstioFederation resource, containing the metadata of the remote cluster, is created on both clusters. The federation is then established automatically...



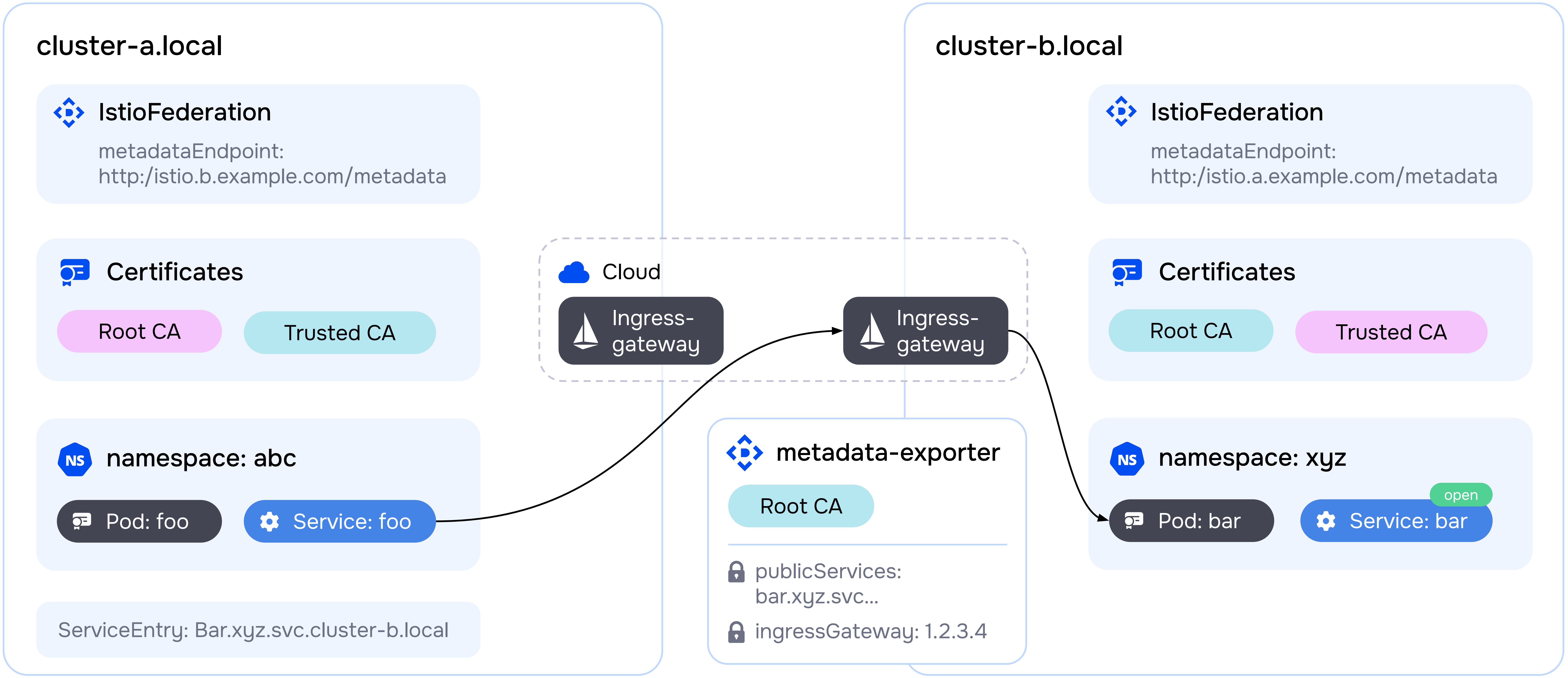
...Deckhouse collects the remote metadata...



...pulls the public root certificate, and exchanges keys to access the private metadata.



Next, it fetches information about the remote cluster's public services and the ingress gateway addresses through which those services are available. Based on this data for each public service, it creates ServiceEntry resources to register remote services on the local cluster.



As a result, the federation gets established, and the public bar.xyz.svc.cluster-b.local service becomes accessible as part of it.