Advanced Application Management Using Red Hat OpenShift Service Mesh

Differences with upstream Istio

Module Topics

- Red Hat OpenShift Service Mesh Upstream
- OpenShift Service Mesh Installation
- Elevated Privileges
- Upstream Istio vs. Maistra
- Maistra

OpenShift Service Mesh Upstream

- OpenShift Service Mesh upstream project: Maistra
    - https://maistra.io
    - https://github.com/Maistra
- Istio upstream:
    - https://istio.io
    - https://github.com/istio
- OpenShift Service Mesh 2.0.x based on Istio 1.6

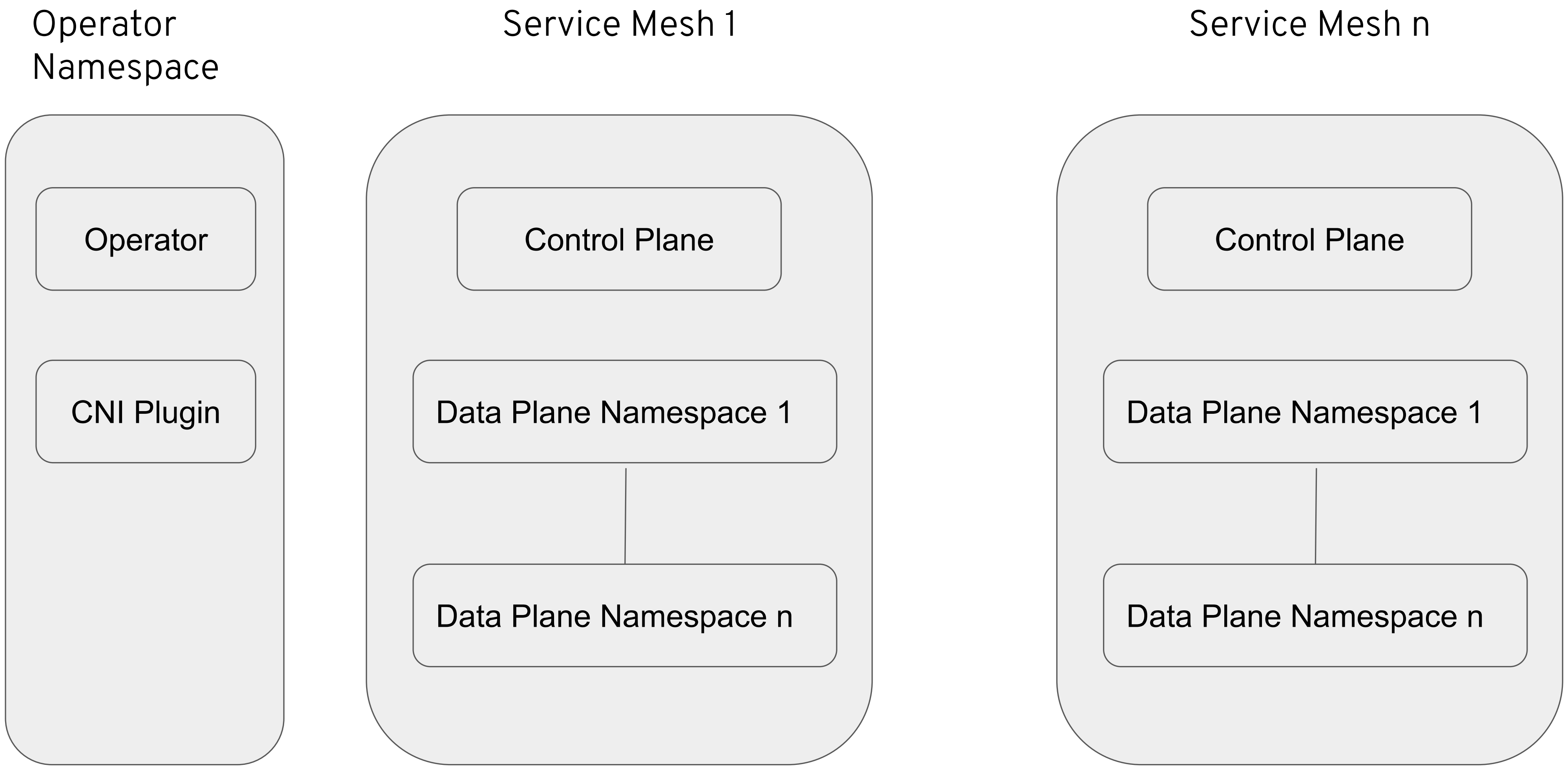
OpenShift Service Mesh Upstream

Goals

- Soft multi-tenancy
  - Support multiple installations
    - Isolated
    - Independent
- Remove need for elevated privileges to install and operate OpenShift Service Mesh instance

OpenShift Service Mesh Upstream

Multi-tenancy



OpenShift Service Mesh Installation

Operator-driven

- Operator uses Helm in background for installation
- Runtime reconciling
- CRDs installed and controlled by operator

OpenShift Service Mesh Installation

OpenShift Service Mesh CRDs

- ServiceMeshControlPlane
  - Defines Service Mesh tenant installation
  - Values defined in CR passed to Helm charts
- ServiceMeshMemberRoll
  - Defines list of member namespaces
  - Control Plane component support modified to support dynamic updates
    - Galley, Pilot, Citadel, Prometheus
  - Use to configure namespaces, SDN

Elevated Permissions

- Proxy init
  - Replaced by Istio CNI (invoked through Multus CNI)
  - Daemon runs in operator namespace
  - Injector adds annotations to pods (partial or full)
- Webhook management
  - Controller in operator
- Cluster role bindings
  - Replaced with role bindings in each member namespace
- Pod locality
  - Controller copies node labels (zone/region) to pod
  - Pilot supports pod labels

Upstream Istio vs. Maistra

Automatic Injection

- Istio upstream
  - Label on namespaces
  - istio-injection: enabled
  - Automatically inject all pods
- Maistra
  - OpenShift Service Mesh namespaces automatically included
  - Pods opt in through annotation
  - sidecar.istio.io/inject: "true"

Upstream Istio vs. Maistra

Cryptographic Library

- Istio upstream
    - Envoy integration with BoringSSL
    - Golang cryptographic library
- Maistra
    - OpenSSL supported by Red Hat
    - OpenSSL FIPS compliance
    - Golang cryptographic library replaced with openssl
    - Envoy integration with OpenSSL

Maistra

Upstream Involvement of Maistra Team

- Release managers for v. 1.3, 1.4, 1.5
- VHDS/incremental xDS
- Upstreaming multi-tenancy work
- Operator development

Module Summary

- Red Hat OpenShift Service Mesh Upstream
- OpenShift Service Mesh Installation
- Elevated Privileges
- Upstream Istio vs. Maistra
- Maistra