Possibilities

#CiscoLive

Benefits on enabling ISTIO on 5G Cloud Native Functions

5G SMF

Solomon Ayyankulankara Kunjan
Technical Leader Services

DGTL-TSCSPG-607

cisco Live!

#CiscoLive

cisco

Agenda

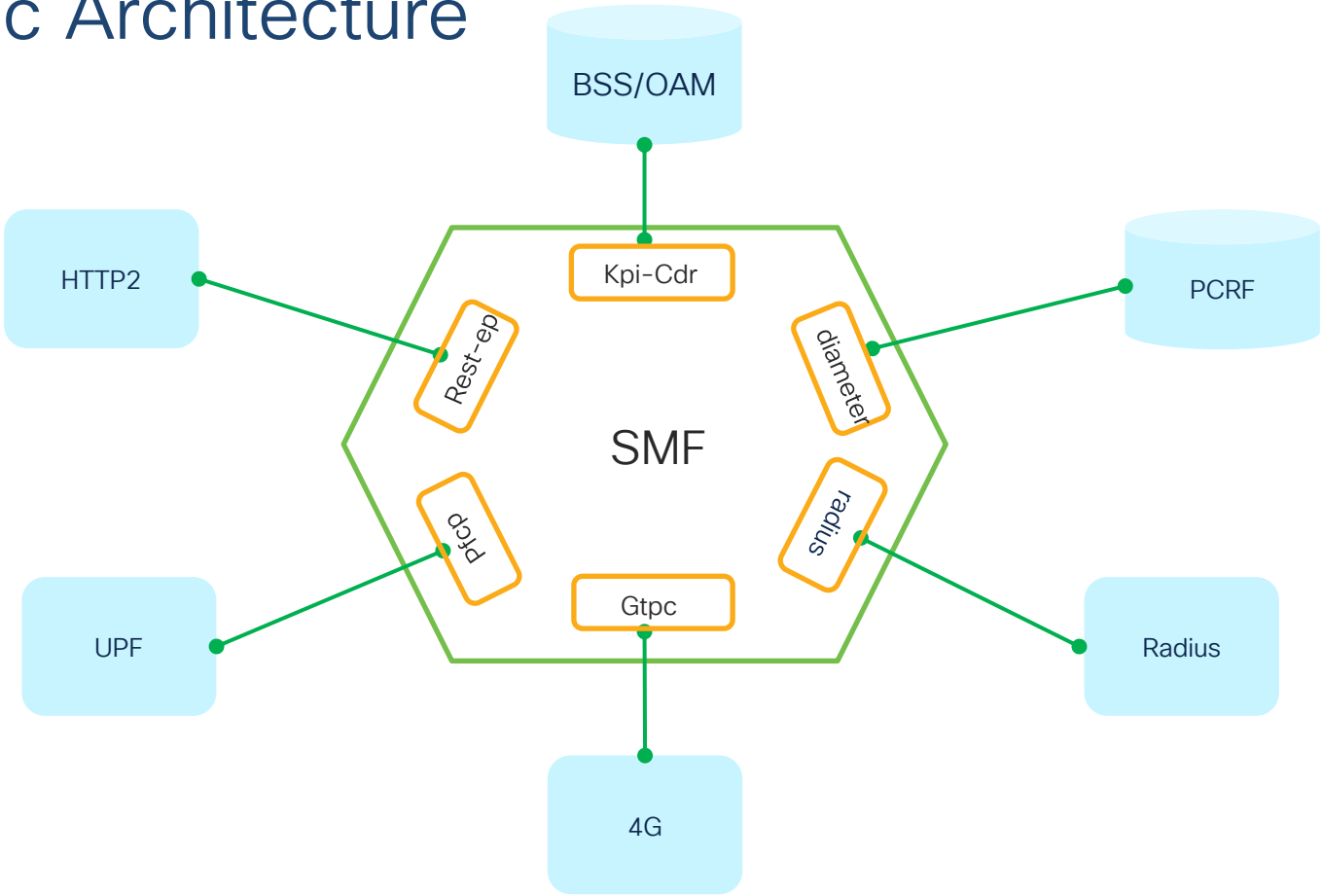
- Introduction
- WHAT is ISTIO
- What is Microservices
- WHY we need ISTIO Service Mesh
- Services of ISTIO
- HOW ISTIO implemented in 5G NF
- Config and CLI Output
- Conclusion

WHAT is ISTIO

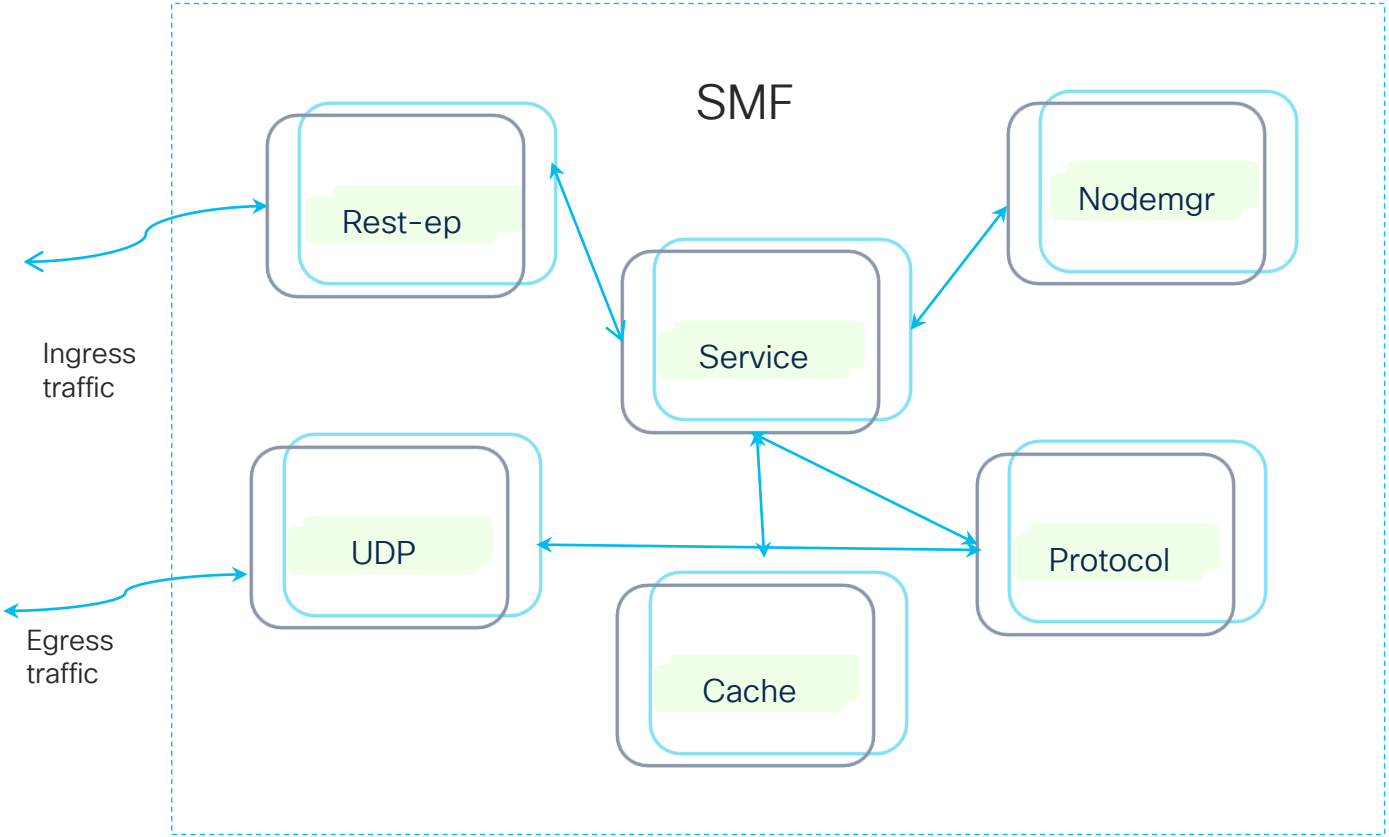
Istio is a Service Mesh

Service Mesh ??

Monolithic Architecture



Microservices



Why Service Mesh

- 3GPP Standard defines 5G NF applications should be in Service based architecture
- This requirement demands all new 5G NF delivered in a Cloud native based solution
- This is the origin for microservice based NF Application in Telco domain
- Cloud native solutions helps Operator / Vendor, test develop and implement the solution

What is Service Mesh

- Service mesh is used to inter-connect mircoservices inside the application

- Service Mesh offers discovery of service, Load balancing , Failure recovery,
- Metrics and Monitoring, Canary Rollouts, Access control and End to End Authentication .

- More the Microservices that exist in the application the greater the challenge is to implement, deploy and operate .

- Istio helps to solve the short coming of kubernetes and other cloud native platforms

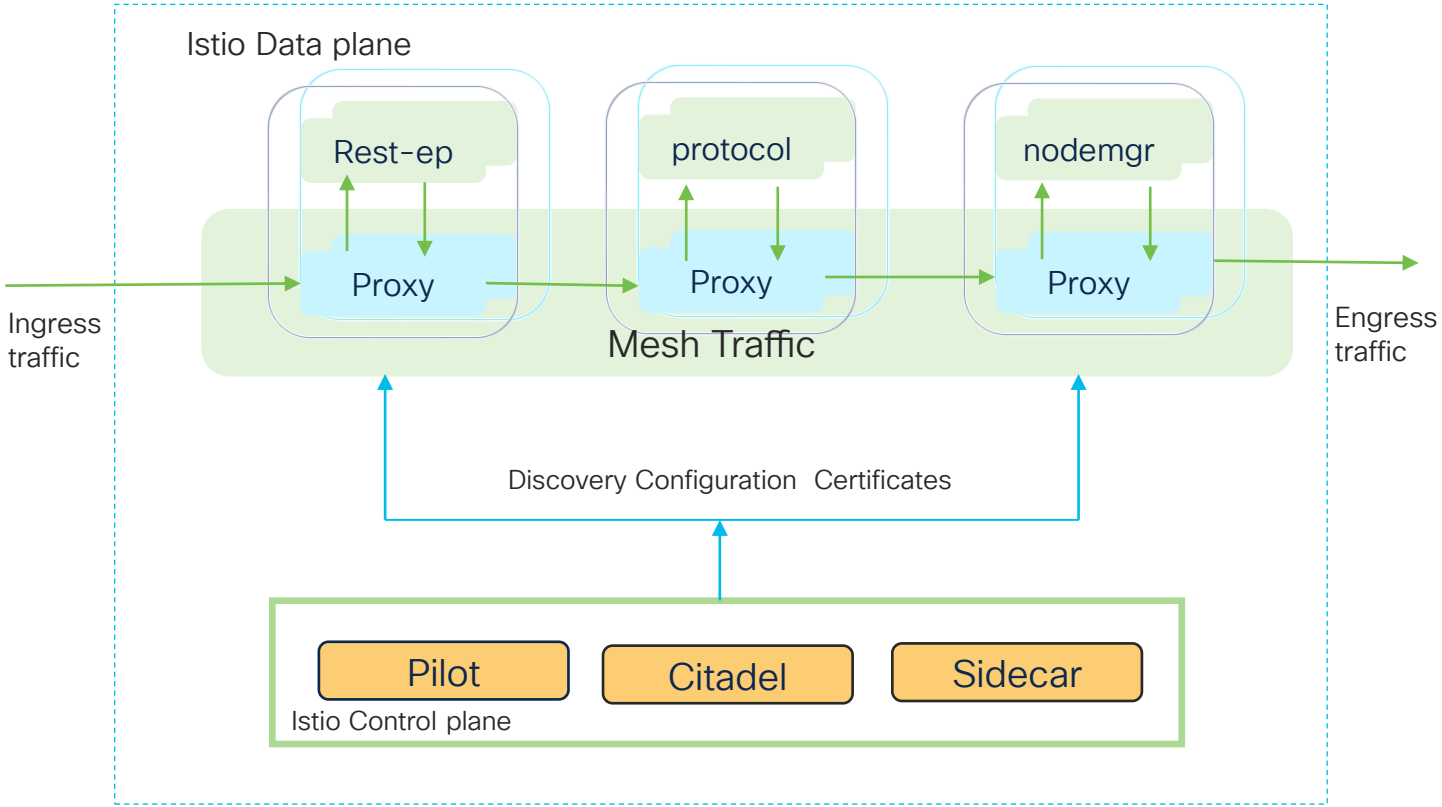
Istio Provides

- Brings visibility and control to Core network
- Automatic load balancing for HTTP, gRPC, WebSocket, and TCP traffic.
- Fine-grained control of traffic behavior with rich routing rules, retries, failovers, and fault injection.
- A pluggable policy layer and configuration API supporting access controls, rate limits and quotas.
- Automatic metrics, logs, and traces for all traffic within a cluster, including cluster ingress and egress.
- Secure service-to-service communication in a cluster with strong identity-based authentication
- and authorization.

Istio Architecture

- *An Istio service mesh is logically split into a **data plane** and a **control plane.***
- The data plane is composed of a set of intelligent proxies (Envoy) deployed as sidecars.
- These proxies mediate and control all network communication between microservices
- The control plane manages and configures the proxies to route traffic.
- The control plane configures and enforce policies in addition to collecting telemetry.

Microservices with Istio



Sidecar

Sidecar injection
sidecar injection is adding the configuration of additional containers to the pod template. The added containers needed for the Istio service mesh are:

istio-init:
This init container is used to setup the iptables rules so that inbound/outbound traffic will go through the sidecar proxy.

An init container is different than an app container in following ways:
- It runs before an app container is started and it always runs to completion.
- If there are many init containers, each should complete with success before the next container is started

istio-init does just that and sets up the iptables rules.
istio-proxy This is the actual sidecar proxy (based on Envoy).

Config

```
[labsmi-cm1] SMI Cluster Deployer(config)# clusters podsmi
[labsmi-cm1] SMI Cluster Deployer(config-clusters-labsmi)# addons istio enabled
[labsmi-cm1] SMI Cluster Deployer(config-clusters-labsmi)# commit
```

```
kubectl get ns --show-labels
NAME              STATUS   AGE     LABELS
cee-global        Active   50d     smi-application=cee
default           Active   51d     <none>
istio-system      Active   51d     name=istio-system
kube-node-lease   Active   51d     <none>
kube-public       Active   51d     <none>
kube-system       Active   51d     <none>
nginx-ingress     Active   51d     name=nginx-ingress
smf-data          Active   4d16h   istio-injection=enabled,smi-application=smf
smf-ims           Active   4d16h   istio-injection=enabled,smi-application=smf
smi-certs         Active   51d     name=smi-certs
smi-vips          Active   51d     name=smi-vips
```

CLI Output

kubectl get all -n istio-system

```
NAME                                        READY   STATUS    RESTARTS   AGE
pod/istio-citadel-59bbc75849-7m72v          1/1     Running   0          51d
pod/istio-pilot-5f897c7c7d-v5ctc            1/1     Running   0          5d12h
pod/istio-sidecar-injector-566954b97d-2k8tm 1/1     Running   0          51d

NAME                               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                 AGE
service/istio-citadel              ClusterIP   10.102.249.135   <none>        8060/TCP,15014/TCP
51d
service/istio-pilot                ClusterIP   10.100.94.207    <none>
15010/TCP,15011/TCP,8080/TCP,15014/TCP   51d
service/istio-sidecar-injector     ClusterIP   10.102.94.196    <none>        443/TCP
51d

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/istio-citadel           3/3     3            3           51d
deployment.apps/istio-pilot             3/3     3            3           51d
deployment.apps/istio-sidecar-injector  3/3     3            3           51d

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/istio-citadel-59bbc75849          3         3         3       51d
replicaset.apps/istio-pilot-567f7cf7b4            0         0         0       51d
replicaset.apps/istio-pilot-5f897c7c7d            3         3         3       5d12h
replicaset.apps/istio-sidecar-injector-566954b97d 3         3         3       51d
```

CLI

```
kubectl describe pod/smf-service-n0-0 -n smf-data
Name:         smf-service-n0-0
Namespace:    smf-data
Priority:     0
Node:         cnat-cnat-core-service-data2/10.192.1.29
Start Time:   Thu, 30 Apr 2020 08:14:16 +0000
Labels:       component=smf-service
              controller-revision-hash=smf-service-n0-6564985bc8
              nID=0
              release=smf-data-smf-service
              statefulset.kubernetes.io/pod-name=smf-service-n0-0
Annotations:  cni.projectcalico.org/podIP: 192.200.6.60/32
              prometheus.io/port: 8080
              prometheus.io/scrape: true
              sidecar.istio.io/status:

{"version":"ef6fc7bbef20d87d3ef458703073b180352c6b8a5d90fe8d387f88d5b4ede2d6","ini
tContainers":["istio-init"],"containers":["istio-proxy"]...
              traffic.sidecar.istio.io/includeInboundPorts: 8805
Status:       Running
IP:           192.200.6.60
IPs:
  IP:           192.200.6.60
Controlled By:  StatefulSet/smf-service-n0
```

```
Init Containers:
  istio-init:
    Container ID:  docker://3c5a51db5b56aa35386bd73ec80ded8aea22919df4b4cc47419f517173e71703
    Image:         dockerhub.cisco.com/smi-fuse-docker-internal/smi-apps/istio/master/proxy_init:1.2.0-
b003e4d
    Image ID:      docker://sha256:df269e5e92ec13a328390716a737b9742523e832ee0d04525018979b4984701a
    Port:          <none>
    Host Port:     <none>
    Args:
      -p
      15001
      -u
      1337
      -m
      REDIRECT
      -i
      10.96.0.0/12
      -x

      -b
      8805
      -d
      15020
    State:          Terminated
      Reason:       Completed
      Exit Code:    0
      Started:      Thu, 30 Apr 2020 08:14:21 +0000
      Finished:     Thu, 30 Apr 2020 08:14:21 +0000
    Ready:          True
```

Load balancing via CLI

```
# show sessions affinity
POD INSTANCE    COUNT
----------------------
smf-rest-ep-0   7038
smf-rest-ep-1   6754
smf-service-0   3392
smf-service-1   3394
smf-service-10  3406
smf-service-11  3421
smf-service-12  3437
smf-service-13  3440
smf-service-2   3378
smf-service-3   3370
smf-service-4   3392
smf-service-5   3386
smf-service-6   3374
smf-service-7   3438
smf-service-8   3428
smf-service-9   3430

# show peers all
```

ENDPOINT	LOCAL ADDRESS	PEER ADDRESS	DIRECTION	POD INSTANCE	TYPE	STATUS	CONNECTED TIME	DISCONNECTED TIME	RPC	ADDITIONAL DETAILS
<none>	192.200.12.197	10.101.26.33:9000	Outbound	smf-rest-ep-1	Rest	Started	37 hours	<none>	AMF	<none>
<none>	192.200.12.197	10.101.27.19:9000	Outbound	smf-rest-ep-1	Rest	Started	37 hours	<none>	AMF	<none>
<none>	192.200.7.178	10.101.26.33:9000	Outbound	smf-rest-ep-0	Rest	Started	37 hours	<none>	AMF	<none>
<none>	192.200.7.178	10.101.101.13:9000	Outbound	smf-rest-ep-0	Rest	Started	30 hours	<none>	AMF	<none>
<none>	192.200.12.197	10.101.24.11:9000	Outbound	smf-rest-ep-1	Rest	Started	38 hours	<none>	AMF	<none>
<none>	192.200.12.197	10.101.24.3:9000	Outbound	smf-rest-ep-1	Rest	Started	38 hours	<none>	AMF	<none>
<none>	192.200.7.178	10.101.27.15:9000	Outbound	smf-rest-ep-0	Rest	Started	37 hours	<none>	AMF	<none>
<none>	192.200.7.178	10.101.26.31:9000	Outbound	smf-rest-ep-0	Rest	Started	38 hours	<none>	AMF	<none>
<none>	192.200.7.178	10.101.21.29:9000	Outbound	smf-rest-ep-0	Rest	Started	38 hours	<none>	AMF	<none>

Thank you

Possibilities

#CiscoLive