# FOCOM Services

## Cloud Registration and Discovery

@startuml "Registration"

Box “Personnel” #lightblue

Actor “Cloud Install \nProject Mgr” as CloudOps

End box

box “FOCOM” <<SMO>> #gold

Database “Cloud Inventory” as CIDB

Boundary “Cloud LCM” as FOCOM

end box

box “SMO Integration Fabric” <<SMO>> #gold

Boundary “Log Store” as Grafana

end box

Box “O-Cloud Platform” #lightseagreen

participant “Genesis Server” as Genesis

Boundary “IMS” as IMS

Boundary “DMS ETSI” as Tacker

Boundary “DMS K8S” as K8S

Boundary “DMS OpenStack” as OpenStack

End box

ref over CloudOps, IMS #royalblue

\*\*<back:royalblue><color: yellow> Reference: WG6 Orchestration and Use Case 3.1.3</color></back>\*\*

End ref

CloudOps -> CIDB : POST ../Clouds

Return “globalCloudId”

Alt ManualRegistration

CloudOps -> FOCOM : POST

Note Right : ../Clouds/Register (globalCloudId, InternalCloudId, IMSendpoint)

FOCOM -> Genesis: Bootstrap

Else Genesis

CloudOps -> Genesis : Bootstrap

Note Right : (globalCloudId, callback=“../Clouds/Register”, Blueprint)

Genesis -> Tacker \*\* : activate

Genesis -> IMS \*\* : activate

IMS -> K8S: discover all k8s clusters

IMS -> IMS: add/update deployment Manager\n representing k8s as O2DMS\n in k8s native API profile

IMS -> Tacker: register k8s as O2DMS\n in ETSI NFV profile

IMS <-- Tacker: VimId

IMS -> IMS: add/update deployment Manager\n representing k8s as O2DMS\n in ESTI NFV profile

IMS -> OpenStack: discover all openstack instances

IMS -> Tacker: register openstack as O2DMS\n in ESTI NFV profile

IMS <-- Tacker: VimId

IMS -> IMS: add/update deployment Manager\n representing openstack as O2DMS\n in ESTI NFV profile

' Genesis -> K8S \*\* : activate

' Genesis -> Tacker : Configure (CISM=K8S, VIM=”K8S/kubevert”)

Genesis -> FOCOM : POST

Note Right : ../Clouds/Register (globalCloudId, InternalCloudId, IMSendpoint)

End if

FOCOM -> CIDB : PATCH

Note Right : ../Clouds/<globalCloudId> (InternalCloudId, IMSendpoint)

FOCOM -> IMS : GET <IMSendpoint>/InfrastructureInventory/v1/ResourceTypes

Loop Foreach ResourceType

FOCOM -> Grafana : POST

Note Right: Log ResourceType

End loop

FOCOM -> IMS : GET <IMSendpoint>/InfrastructureInventory/v1/Resources

Loop Foreach Resource

FOCOM -> Grafana : POST

Note Right: Log Resource

End loop

FOCOM -> IMS : GET <IMSendpoint>/InfrastructureInventory/v1/ResourcePools

Loop Foreach ResourcePool

FOCOM -> Grafana : POST

Note Right: Log ResourcePool

End loop

FOCOM -> IMS : GET <IMSendpoint>/o2ims-infrastructureInventory/v1/deploymentManagers

Note over FOCOM, IMS

DMS Attributes

ProfileType: K8S, K8s API endpoint, Capabilities={K8S,...}, Capacity=1, Location=us-central1

ProfileType: K8S, K8s API endpoint, Capabilities={K8S, Low Latency,...}, Capacity=2, Location=us-edge1

ProfileType: ESTI\_NFV, Tacker API endpoint, Capabilities={ETSI, K8S}, Capacity=1,Location=us-central1

ProfileType: ESTI\_NFV, Tacker API endpoint, Capabilities={ETSI, K8S, Low Latency}, Capacity=2,Location=us-edge1

ProfileType: ESTI\_NFV, Tacker API endpoint, Capabilities={ETSI, OpenStack, Low Latency}, Capacity=3,Location=us-edge1

End note

Return

Loop Foreach discovered DMS

FOCOM -> Grafana : POST

Note Right: Log DMS

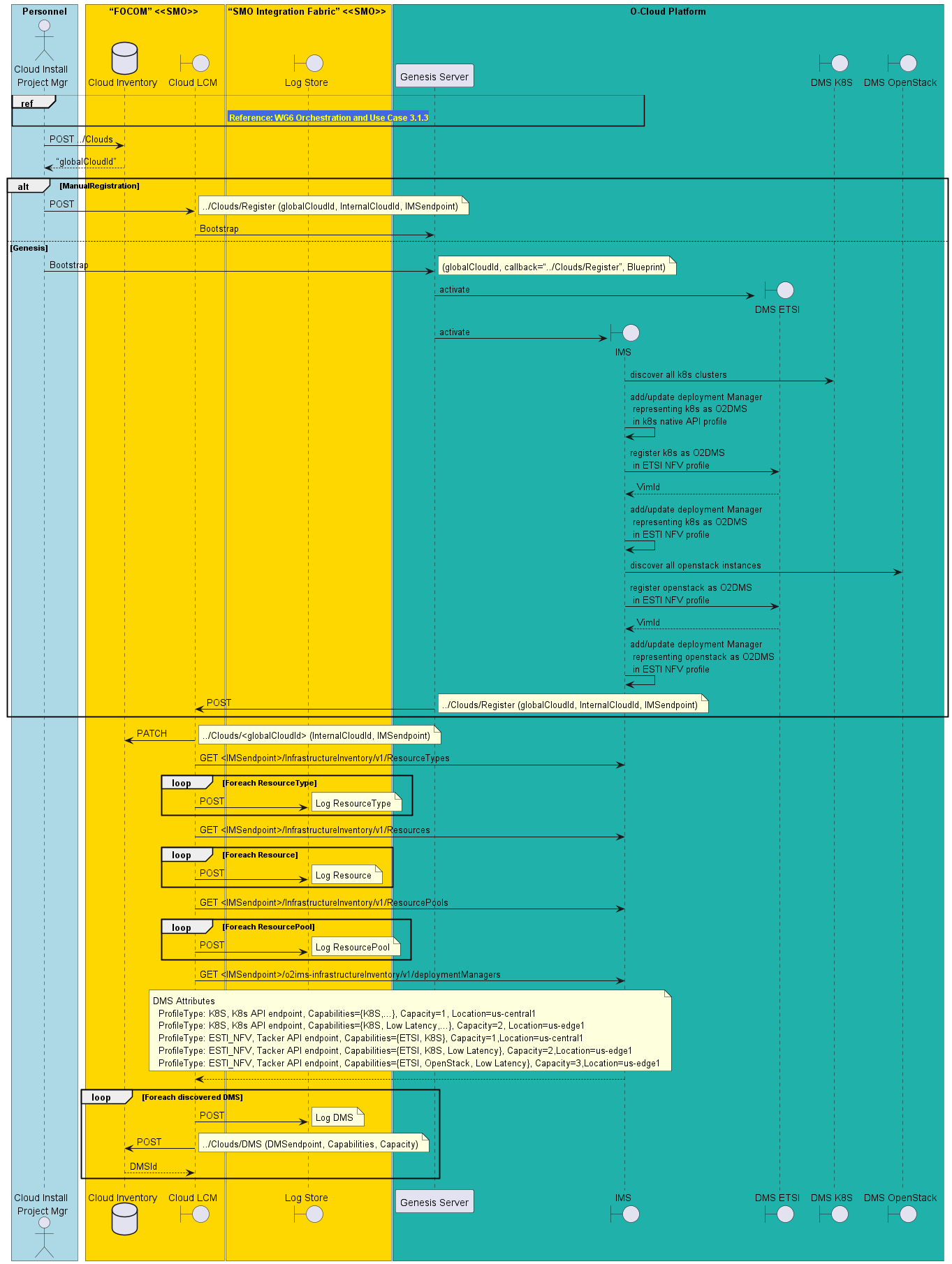
FOCOM -> CIDB : POST

Note Right : ../Clouds/DMS (DMSendpoint, Capabilities, Capacity)

Return DMSId

End loop

@enduml



## Homing

@startuml

box “FOCOM” <<SMO>> #gold

Database “Cloud Inventory” as CIDB

Boundary “Cloud LCM” as FOCOM

Boundary “Homing” as HOMING

end box

box “NFO” <<SMO>> #gold

Database “Deployment Inventory” as AppDB

Boundary “NF LCM” as NFO

end box

NFO -> NFO

Note Left : \*\*Create Homing Demands:\*\*\nforeach deployment item\n\tidentify LocationIdentifier\n\tidentify ProfileType\nend loop

NFO -> HOMING : Create Homing Plan

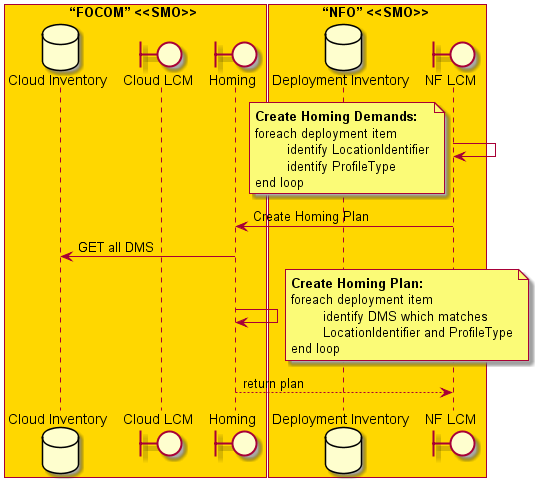
HOMING -> CIDB : GET all DMS

HOMING -> HOMING

Note Right : \*\*Create Homing Plan:\*\*\nforeach deployment item\n\tidentify DMS which matches\n\tLocationIdentifier and ProfileType\nend loop

HOMING --> NFO : return plan

@enduml



## Orchestration

@startuml "Orchestration"

box “FOCOM” <<SMO>> #gold

Database “Cloud Inventory” as CIDB

Boundary “Cloud LCM” as FOCOM

end box

box “NFO” <<SMO>> #gold

Database “Deployment Inventory” as AppDB

Boundary “NF LCM” as NFO

end box

Box “O-Cloud Platform” #lightseagreen

Boundary “DMS ETSI” as Tacker

Boundary “DMS K8S” as K8S

Boundary “DMS OpenStack” as OpenStack

End box

NFO -> NFO: orchestrate workload by homing plan

NFO -> CIDB : Get O2DMS by DMSid

NFO <-- CIDB: O2DMS API endpoint\n (either DMS K8S API endpoint or DMS ESTI API endpint)

Alt DMS in K8S Profile

NFO -> K8S: deploy workload via K8S APIs

NFO -> K8S: query workload resources

NFO -> AppDB: update inventory

ELSE DMS in ESTI Profile

NFO -> Tacker: deploy workload via ETSI NFV APIs

Alt k8s workload

Tacker -> K8S: deploy workload via K8S APIs

ELSE OpenStack workload

Tacker -> OpenStack: deploy workload via OpenStack APIs

End if

NFO -> Tacker: query workload resources

NFO -> Tacker: update inventory

End if

@enduml



@startuml

Box “Personnel” #lightblue

Actor “Cloud Install \nProject Mgr” as CloudOps

Actor “NF Install \nProject Mgr” as NFOps

End box

box “FOCOM” <<SMO>> #gold

Database “Cloud Inventory” as CIDB

Boundary “Cloud LCM” as FOCOM

Boundary “Homing” as HOMING

end box

box “NFO” <<SMO>> #gold

Database “Deployment Inventory” as AppDB

Boundary “NF LCM” as NFO

end box

box “OAM” <<SMO>> #gold

Boundary “VES Collector” as VES

Boundary “NETCONF Client” as SDNR

end box

Box “RAN PM” <<SMO>> #gold

Database “Telemetry Store” as InfluxDB

End box

Box “RAN FM” <<SMO>> #gold

Database “Event Store” as AlarmDB

End box

Box “RAN CM” <<SMO>> #gold

Database “Config Persistent Store” as CPS

End box

box “SMO Integration Fabric” <<SMO>> #gold

Boundary “Message Bus” as Kafka

Boundary “Log Store” as Grafana

end box

Box “O-Cloud Platform” #lightseagreen

participant “Genesis Server” as Genesis

Boundary “DMS ETSI” as Tacker

Boundary “DMS K8S” as K8S

Boundary “IMS” as IMS

End box

Box “O-RAN” #lightpink

Participant “CNF” as CNF

Participant “CNF\nComponent” as CNFC

end box

NFO -> NFO

Note Right : \*\*Create Homing Demands:\*\*\nforeach deployment item\n\tidentify LocationIdentifier\n\tidentify ProfileType\nend loop

NFO -> FOCOM : Create Homing Plan

FOCOM -> CIDB : GET all DMS

FOCOM -> FOCOM

Note Right : \*\*Create Homing Plan:\*\*\nforeach deployment item\n\tidentify DMS which matches LocationIdentifier and ProfileType\nend loop

FOCOM --> NFO : return plan

@enduml

