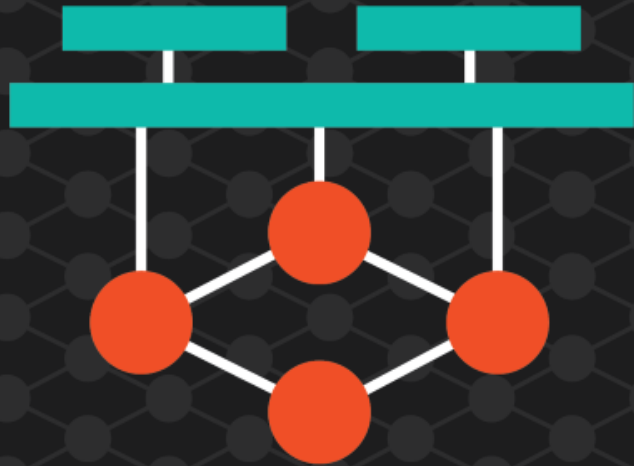


March 26-29, 2018
Los Angeles, CA



ons

NORTH AMERICA

OPEN NETWORKING //

Integrate, Automate, Accelerate



ons
NORTH AMERICA
OPEN NETWORKING //
Integrate, Automate, Accelerate

March 26-29, 2018
Los Angeles, CA

Using Microservices Architecture and Patterns to address application requirements on MEC

Presented by Cloud Architects: Prem Sankar Gopannan (E///) and Prakash Ramchandran (DellEMC)



ons
NORTH AMERICA
OPEN NETWORKING //
Integrate, Automate, Accelerate

Agenda – Part 1 – Prem Sankar G, Ericsson

- MEC Application requirements
- Microservices architecture and patterns
- Walkthrough – Sample app



ons
NORTH AMERICA
OPEN NETWORKING //
Integrate, Automate, Accelerate

Application Types

- Offline applications
 - Fat Client
 - Processing done locally at UE
 - Data feed from cloud
- Online Applications
 - Presentation layer
 - All processing done online
 - Requires connectivity



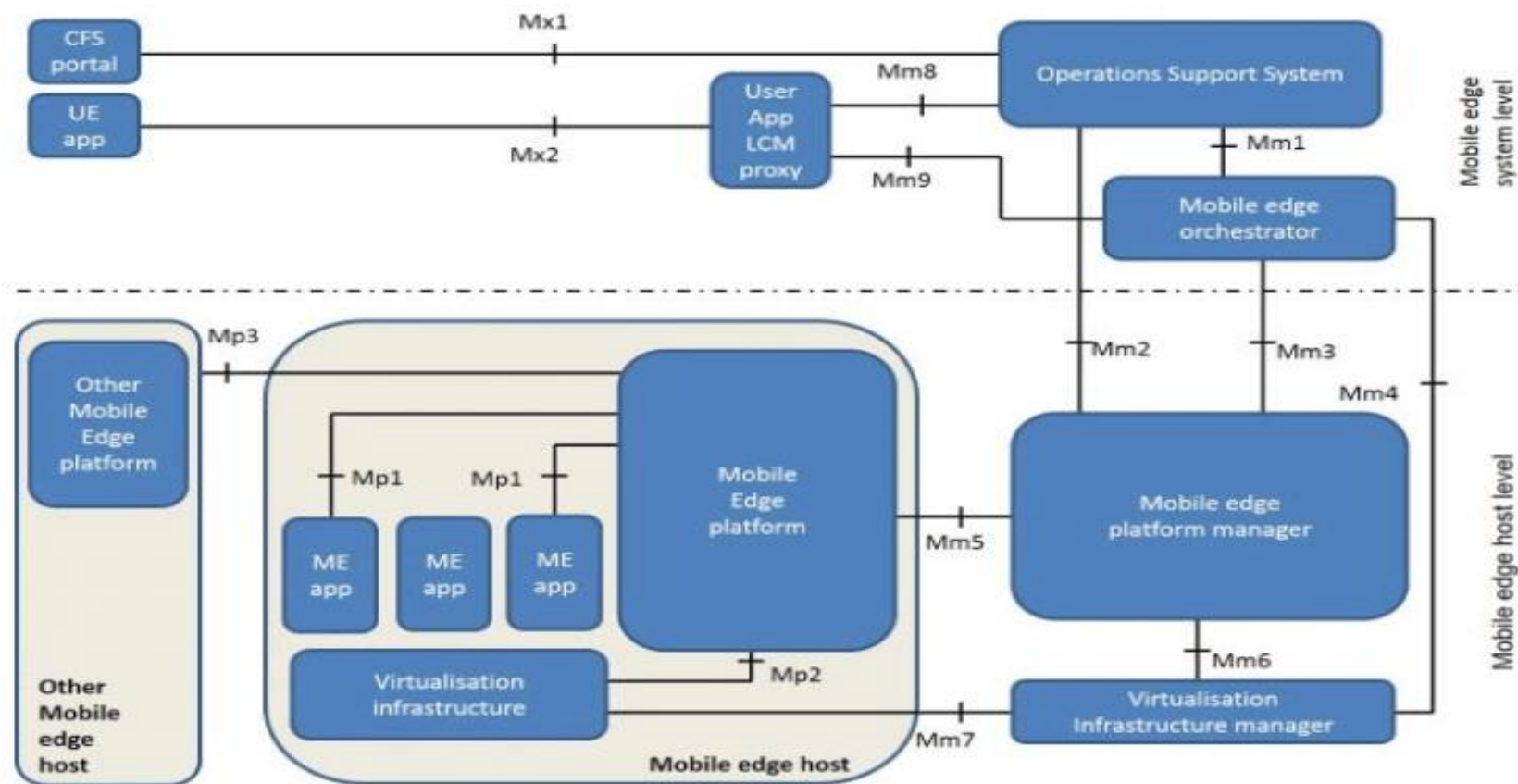
ons
NORTH AMERICA
OPEN NETWORKING //
Integrate, Automate, Accelerate

Key requirements for Applications

- Bandwidth
- Latency, Jitter and Delay
- Location and Presence



MEC Architecture



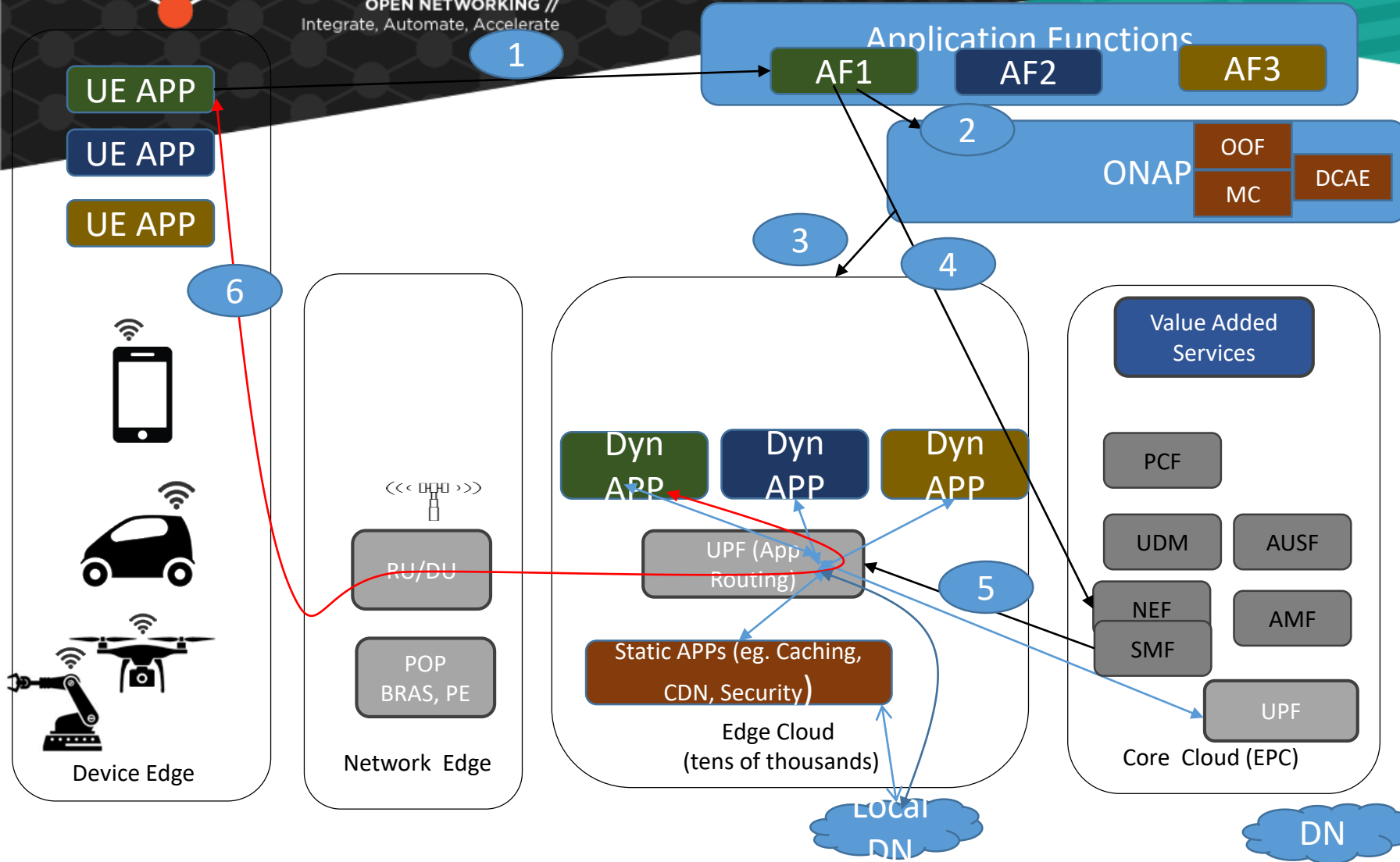
Network E2E with Edge Apps – An example Traffic flow



ONS

NORTH AMERICA

OPEN NETWORKING //
Integrate, Automate, Accelerate



1. UE APP requests AF to create compute for it (Eg. AR/VR, Gaming offload). UE provides information such as 'coordinates', 'max cost', 'max latency' etc...
2. AF requests ONAP to create VNFs in edge that is closer to UE that satisfies UE request(ONAP figures out the best region out of thousands of clouds)
3. ONAP brings up VNFs on the Edge Cloud using VIM API
4. AF informs NEF/SMF to create traffic rule (to enable redirection of UE APP traffic to new VNFs created).
5. SMF informs UPF in the edge cloud (Programs UE classifier of UPF).
6. When the traffic comes from UE APP, right application in the edge cloud receives the traffic.



ons
NORTH AMERICA
OPEN NETWORKING //
Integrate, Automate, Accelerate

Sample application – Pothole fixer



- **Dashboard app** – Control app for the equipment that connects with edge network/Dispatcher app
- **Dispatcher app** – Dispatches jobs to equipment
- **Potholefix app** – App that is in core cloud has info about pothole, location and other information

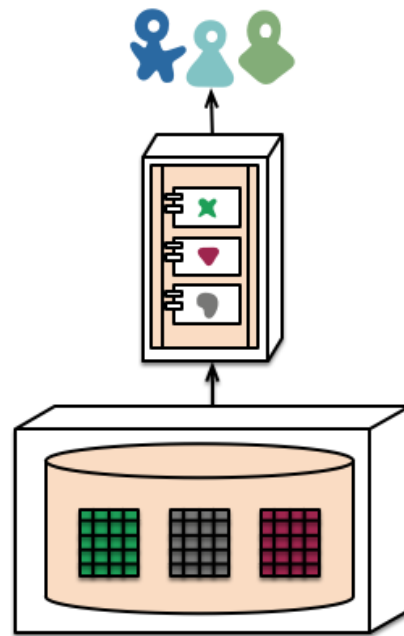


Sample application – Pothole fixer

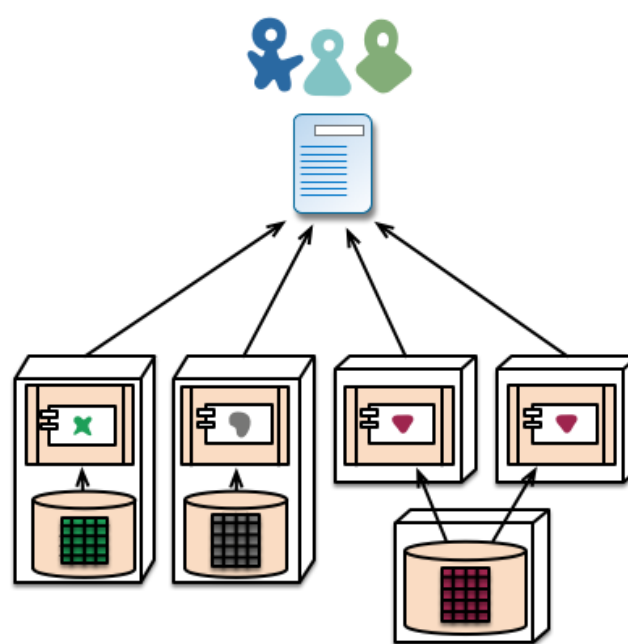
- System functions
 1. AF Request
 2. Select VIM and triggers VNF request
 3. Bring up VNF on the selected edge and establish transport path
 4. Redirect job to payload
- Application functions
 1. Require Fix – Pothole Image, City Authority (for budget)
 2. Response to bobcat dashboard application
 - ...
 3. Terminate application



Microservice Architecture – Quick Intro

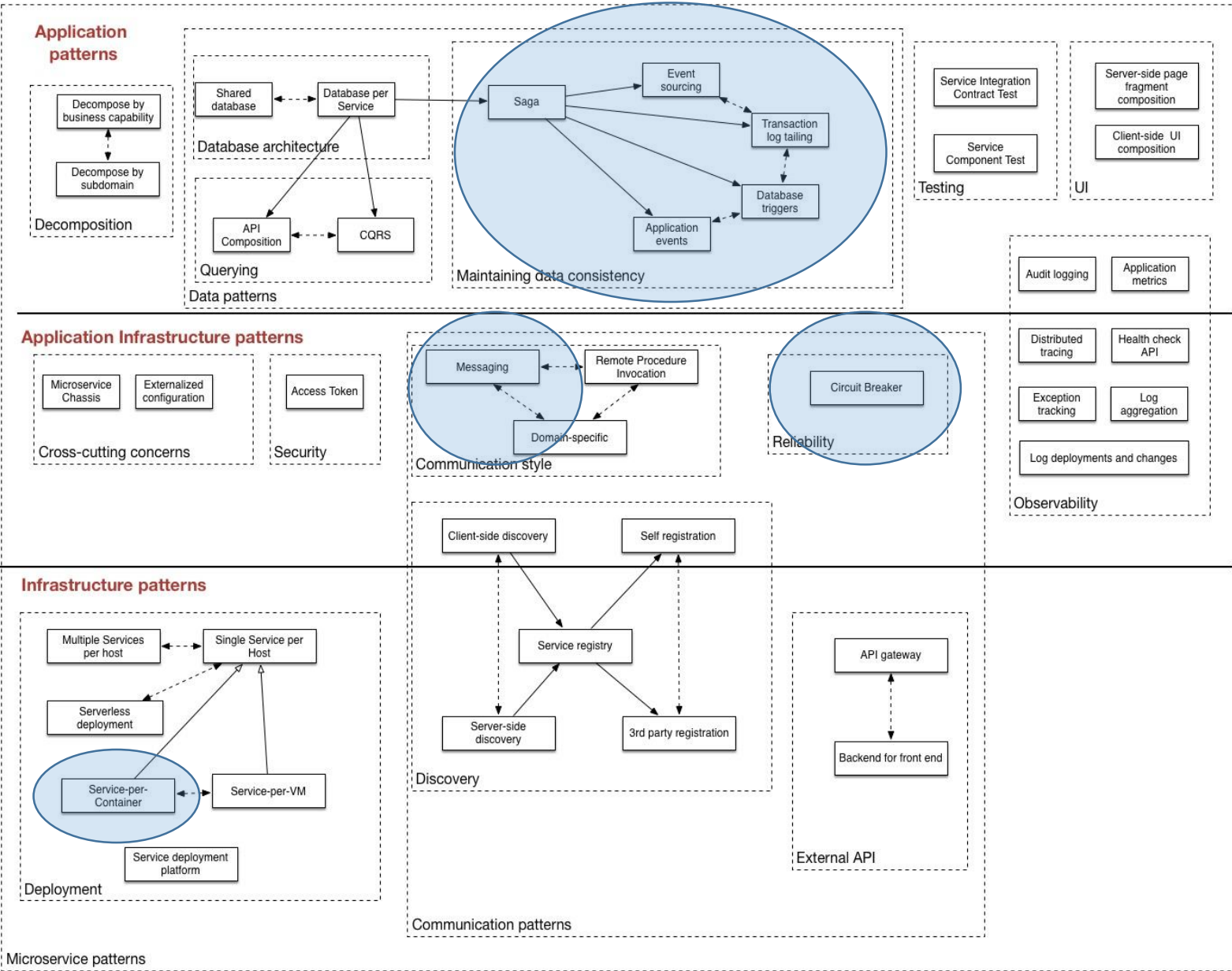
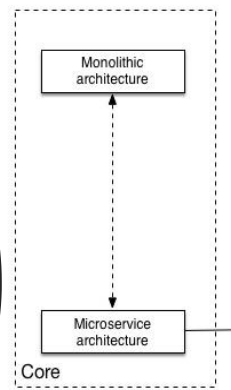
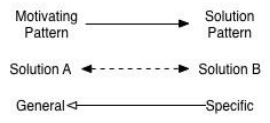


monolith - single database



microservices - application databases

Microservices Design Patterns





ons
NORTH AMERICA
OPEN NETWORKING //
Integrate, Automate, Accelerate

SAGA Patterns

- Database per Service
- Replaces 2 Phase Commit
- Uses Event streams to co-ordinate between the distributed database
- Co-ordination via two ways
 - Choreography - each local transaction publishes domain events that trigger local transactions in other services
 - Orchestration - an orchestrator (object) tells the participants what local transactions to execute

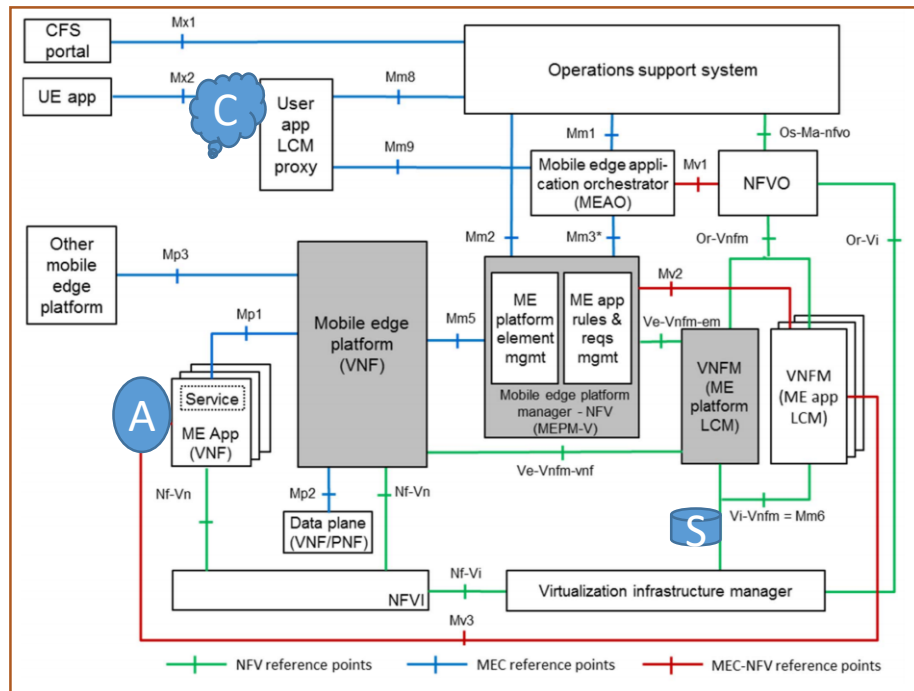


Agenda – Part 2 by R.Prakash (Openstack)

- MEC – Architecture
- Cloudlet and Statelet
- Statelet Design and Flow
- Statelet proposed APIs
 - Admin API for Persistent Volume (PV)
 - User API for Persistent Volume (PV)
- Deployment options – OpenStack Zun micro services for PV



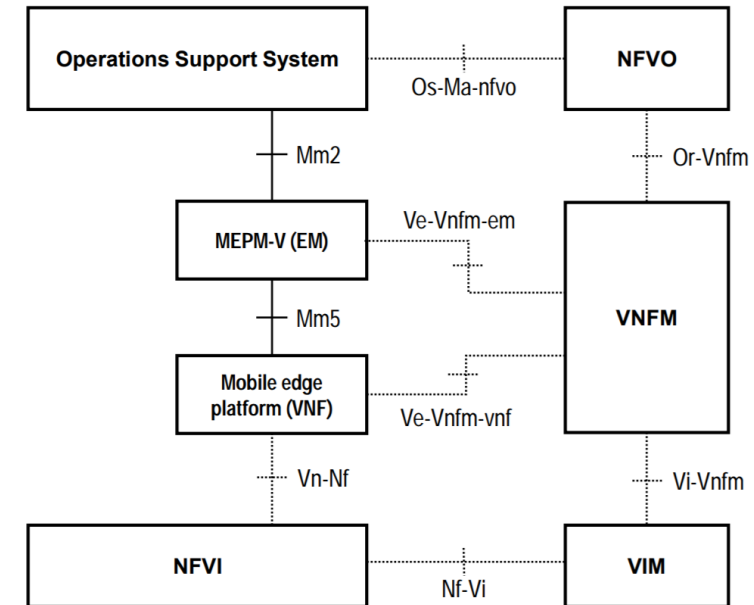
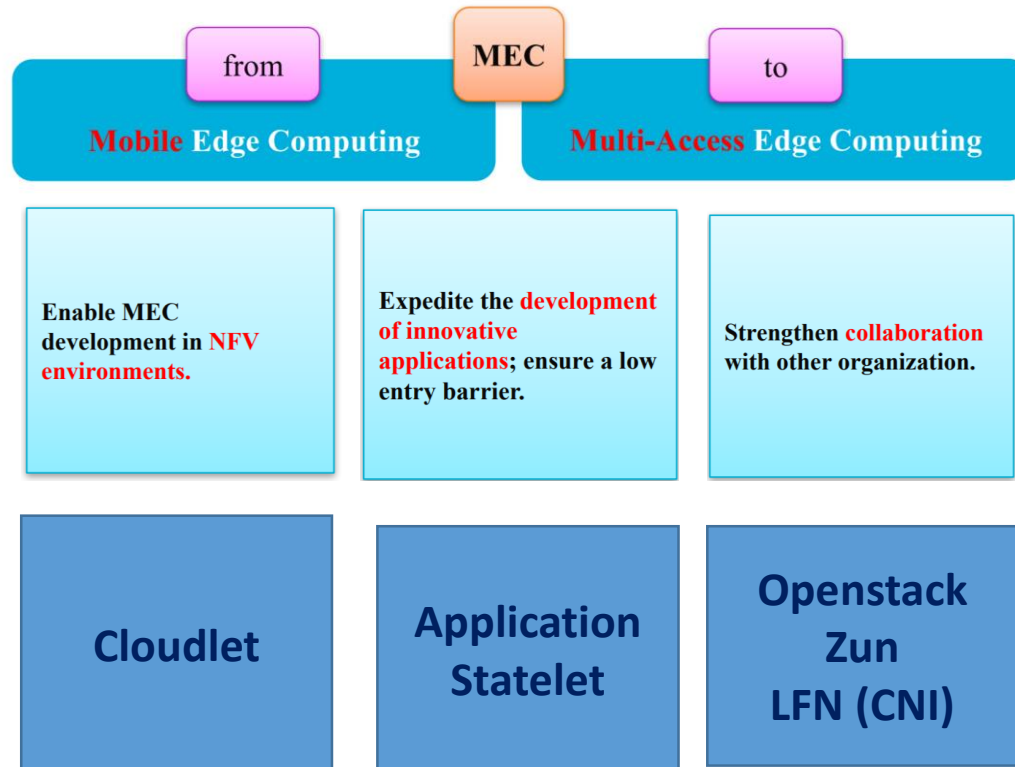
MEC reference architecture in a NFV environment



- UE uses **Cloudlet(C)** API over Mx2: UE App to User/App Proxy (Ambassador Pattern)
- MEPM manages MEP over Mm5
- MEP manages **(A) App VNF** over Mp1
- **Statelet(S)** is a state management service to MEP & ME App VNF LCM over Mm6/Vi-Vnm



Mobile Edge Computing Standard V1 to V2



Management of the ME platform as a VNF



ons
NORTH AMERICA
OPEN NETWORKING //
Integrate, Automate, Accelerate

State Management for Edge Cloud on Server Side

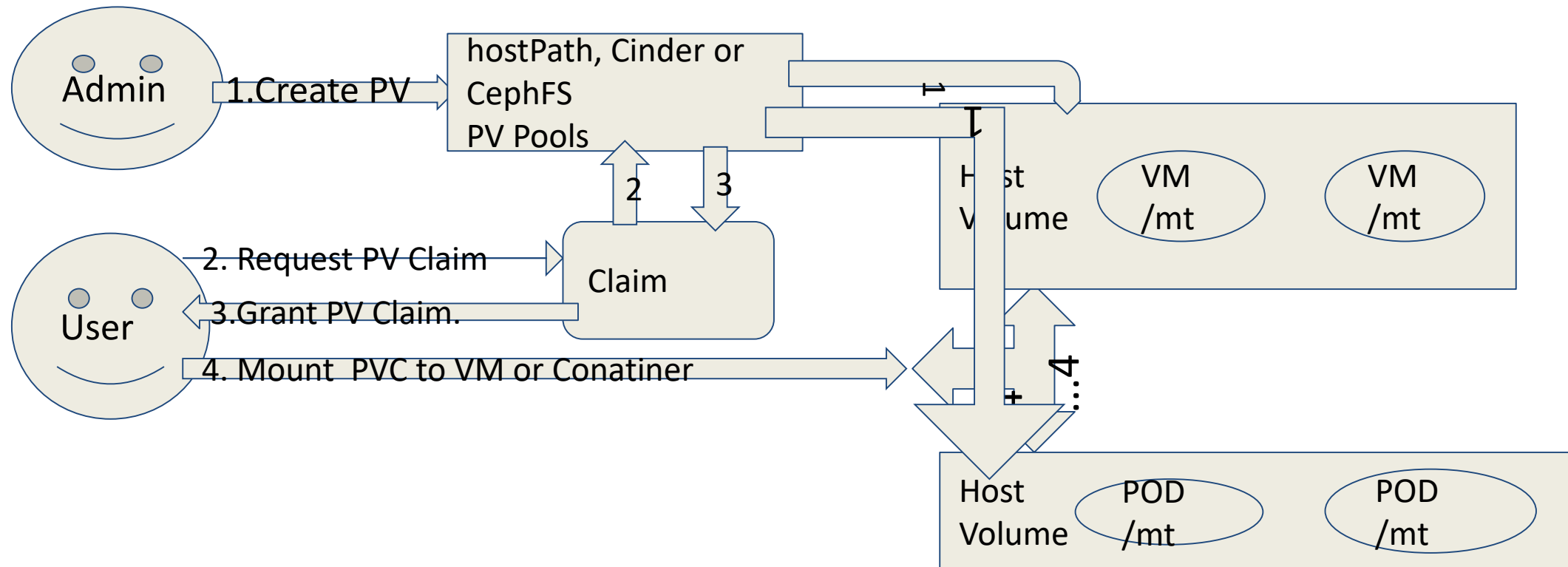
UE talks to Cloudlet VM or container through Proxy for running Application

MEP Manager manages MEP
MEP manages Application

Statelet Support Cloudlet and Application to manage Application State using Shared Persistent Volume pools with mount and unmount calls.



Persistent Volume for state management at Edge





State Management using Persistent Volume for Edge Cloud

- A Persistent Volume (PV) is a network attached storage in the cluster, which is provisioned by the administrator and user claims them from the Pools to mount them in PODs or VMs and finally used by the applications contained in them.
- PVs based on the Storage Class resource. A StorageClass contains pre-defined provisioners and parameters to create a Persistent Volume.
- Volume Types that support managing storage using Persistent Volumes are several but we will limit it to hostPath, CephFS or Cinder as in Openstack for Cloudlet. For minikube demo we use hostPath. Note for VMs use Cinder/CephS.
- So the Object PV will have C,RC,GC,MT(attach), UMT(detach) REST Calls for Cloudlets to support State management for both Openstack VMs and Containers.



Admin APIs

PV Pool, PV/PVExtend

Object	Action	Type	Attributes
PV_Pool	Create	Default	PV of Type hostPath
PV_Pool	Delete	Default	PV of Type hostPath
PV	Create	hostPath /Default	Adapt from k8s
PV	Create	Cinder / BlockStore	Adapt from Openstack
PV	Create	CephFS/ FileStore	Adpat from Ceph
PVExtend	Create	Host Volume	k8s Worker Node
PVExtend	Create	Host Volume	Openstack compute Node
PV	Delete	PV Name	PV_Pool
PV_Pool	Delete	PV_Pool Name	PV list



User APIs PV Claim, Grant, Mount, UnMount

Object	Action	Type	Attributes
PV	Claim	PV_Pool	Capacity
PV	Grant	PV_Pool	Capacity
PV	Mount	PV Name	Mount Device
PV	UnMount	Mount Device	PV Name
PV	Assign	PV Name	PV_Pool
PV_Pool	Assign	PV_Pool Name	PV list



References

MEC: Cloudlet, Statelet, ME Applications

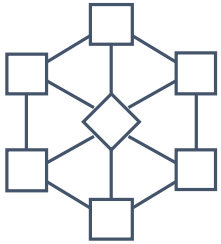
- <http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing>
- http://www.etsi.org/deliver/etsi_gr/MEC/001_099/017/01.01.01_60/gr_MEC017v010101p.pdf
- <https://www.cncf.io/blog/2017/05/23/cncf-hosts-container-networking-interface-cni/>
- <https://docs.openstack.org/queens/api/>
- Refer APIs for Zun, Senlin, manila to define and design Statelet
- First add Statelet API for PV (Admin/User) to Zun for Containers
- Use Senlin to form container clusters using Zun and use Senlin Profile and Policy for placement
- Use Shared File Systems service (manila) if you use CephFS
- <https://www.openstack.org/edge-computing/>
- <https://www.openstack.org/assets/edge/slides/2017-09-07-a-satya-opendev-keynote.pdf>
- <https://docs.openstack.org/kolla/latest/>
- <https://wiki.opnfv.org>
- <https://wiki.onap.org>
- <http://microservices.io/>
- Contacts: Prem Sanka Gopannan (premsankar@gmail.com)/ Prakash Ramchandran (cloud24x7@ieee.org)



ons
NORTH AMERICA

Solutions and Ecosystem

Advanced Architecture



SD-WAN,
vCPE/uCPE

NFV, SDN

SAS, Big Data,
Real-Time Visibility

CORD, MEC, MDC

C-RAN, CUPS,
Network Slicing

