



ONAP & MEC

Leverage ONAP to build MEC Management Layer

Authors: Gaurav Agrawal(Huawei), Chengli Wang (CMCC)

Index

- ❑ MEC Introduction
- ❑ MEC Architecture
- ❑ Reference Implementation
using ONAP

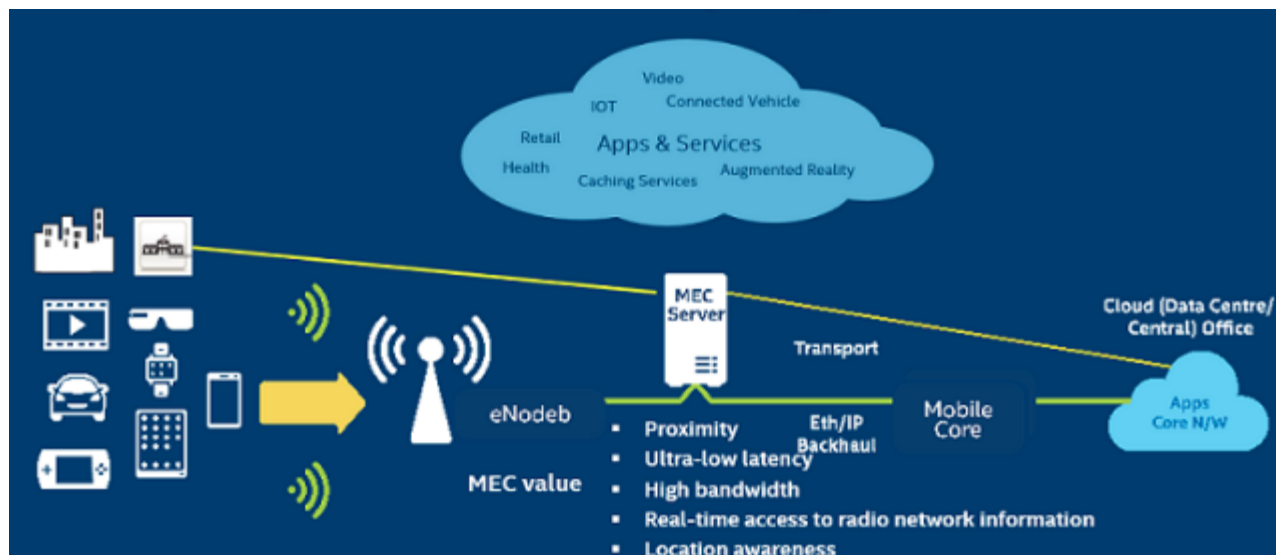




MEC Introduction

Multi Access Edge Computing

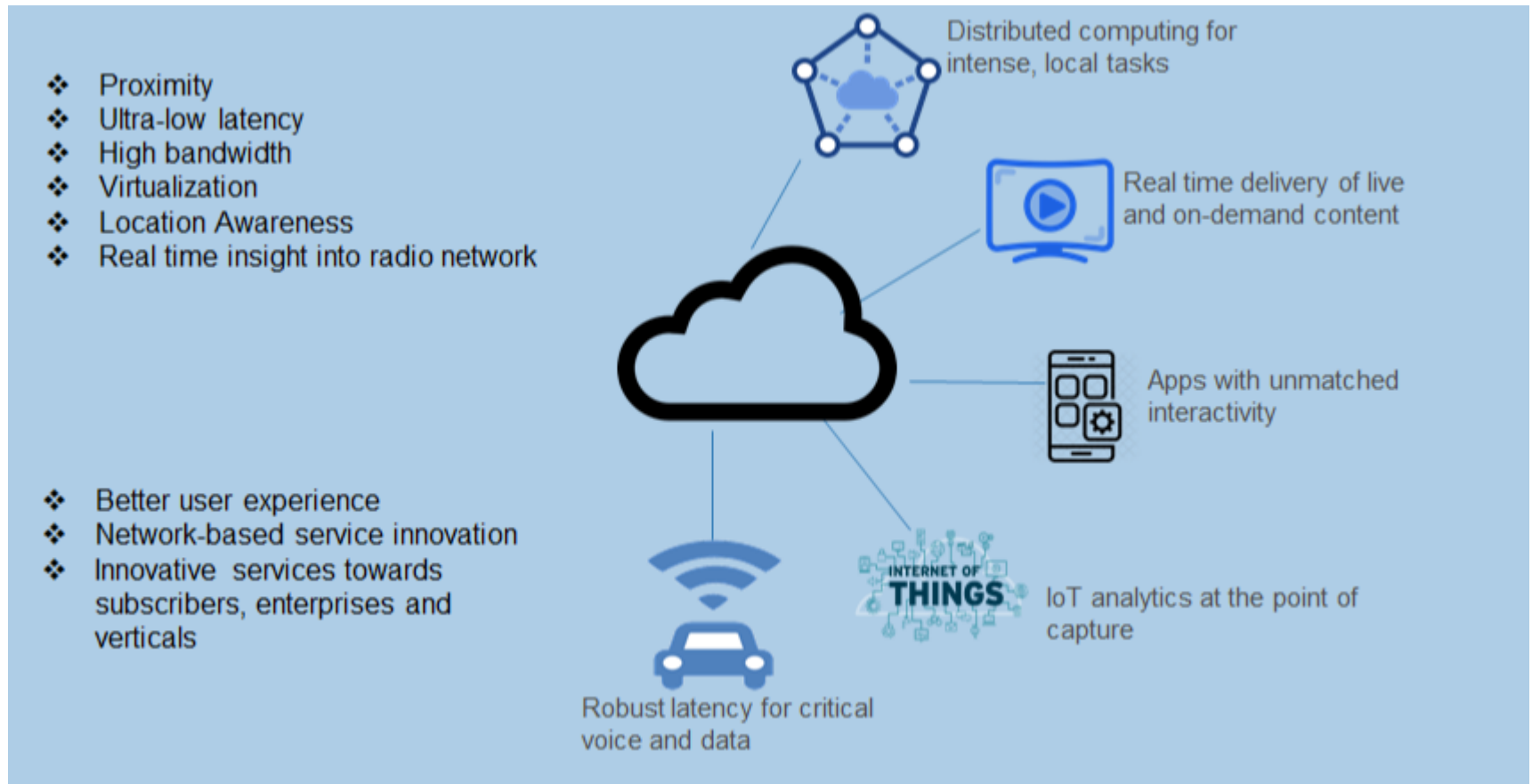
Multi-access Edge Computing (MEC) offers application developers and content providers cloud-computing capabilities and an IT service environment at the edge of the network. This environment is characterized by ultra-low latency and high bandwidth as well as real-time access to radio network information that can be leveraged by applications.



MEC provides a new ecosystem and value chain. Operators can open their Radio Access Network (RAN) edge to authorized third-parties, allowing them to flexibly and rapidly deploy innovative applications and services towards mobile subscribers, enterprises and vertical segments.

MEC Role & Activities The Multi-access Edge Computing (MEC) initiative is an Industry Specification Group (ISG) within ETSI. The purpose of the ISG is to create a standardized, open environment which will allow the efficient and seamless integration of applications from vendors, service providers, and third-parties across multi-vendor Multi-access Edge Computing platforms.

MEC Use Cases



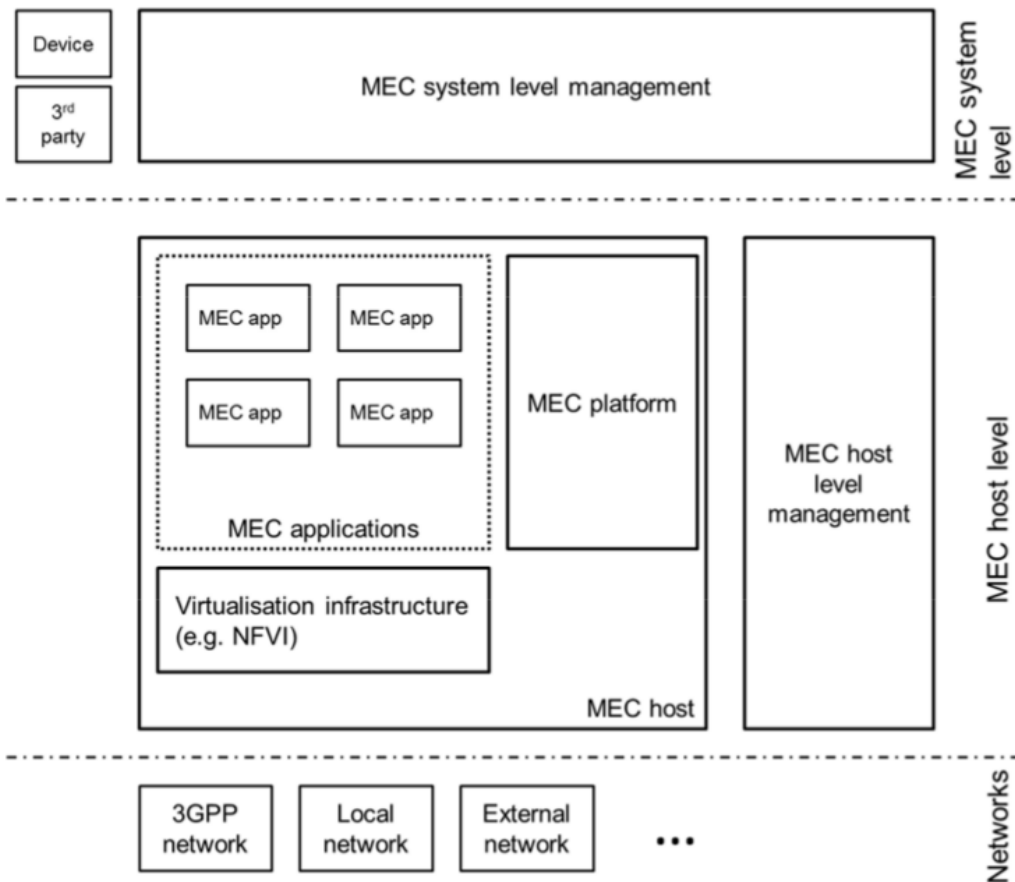
MEC is a natural development in the evolution of mobile base stations and the convergence of IT and telecommunications networking. Multi-access Edge Computing will enable new vertical business segments and services for consumers and enterprise customers. Use cases include:

- ❖ video analytics
- ❖ location services
- ❖ Internet-of-Things (IoT)
- ❖ augmented reality
- ❖ optimized local content distribution and
- ❖ data caching



MEC Architecture

MEC Reference Architecture



The **MEC host** is an entity that contains a MEC platform and a virtualization infrastructure which provides compute, storage, and network resources, for the purpose of running MEC applications.

The **MEC platform** is the collection of essential functionality required to run MEC applications on a particular virtualization infrastructure and enable them to provide and consume MEC services. The MEC platform can also provide services.

MEC applications are instantiated on the virtualization infrastructure of the MEC host based on configuration or requests validated by the MEC management.

The **MEC management (MECM)** comprises the **MEC system level management** and the **MEC host level management**.

The **MEC system level management** includes the **Multi-access edge orchestrator** as its core component, which has an overview of the complete MEC system.

The **MEC host level management** comprises the **MEC platform manager** and the **virtualization infrastructure manager**, and handles the management of the MEC specific functionality of a particular MEC host and the applications running on it.

OSS

- receives requests via the CFS portal and from device applications for instantiation or termination of applications
- Granted requests are forwarded to the multi-access edge orchestrator for further processing.
- receives requests from device applications for relocating applications between external clouds and the MEC system.

User APP LCM Proxy

- allows device applications to request on-boarding, instantiation, termination of user applications
- relocation of user applications in and out of the MEC system.
- Informs the device applications about the state of the user applications.
- Authorizes requests from device applications in the device and interacts with the OSS and the multi-access edge orchestrator for further processing of these requests.

VIM

- Allocating, managing and releasing virtualized resources
- preparing the virtualization infrastructure to run a software image.
- collecting and reporting performance and fault information about the virtualized resources;
- when supported, performing application relocation.

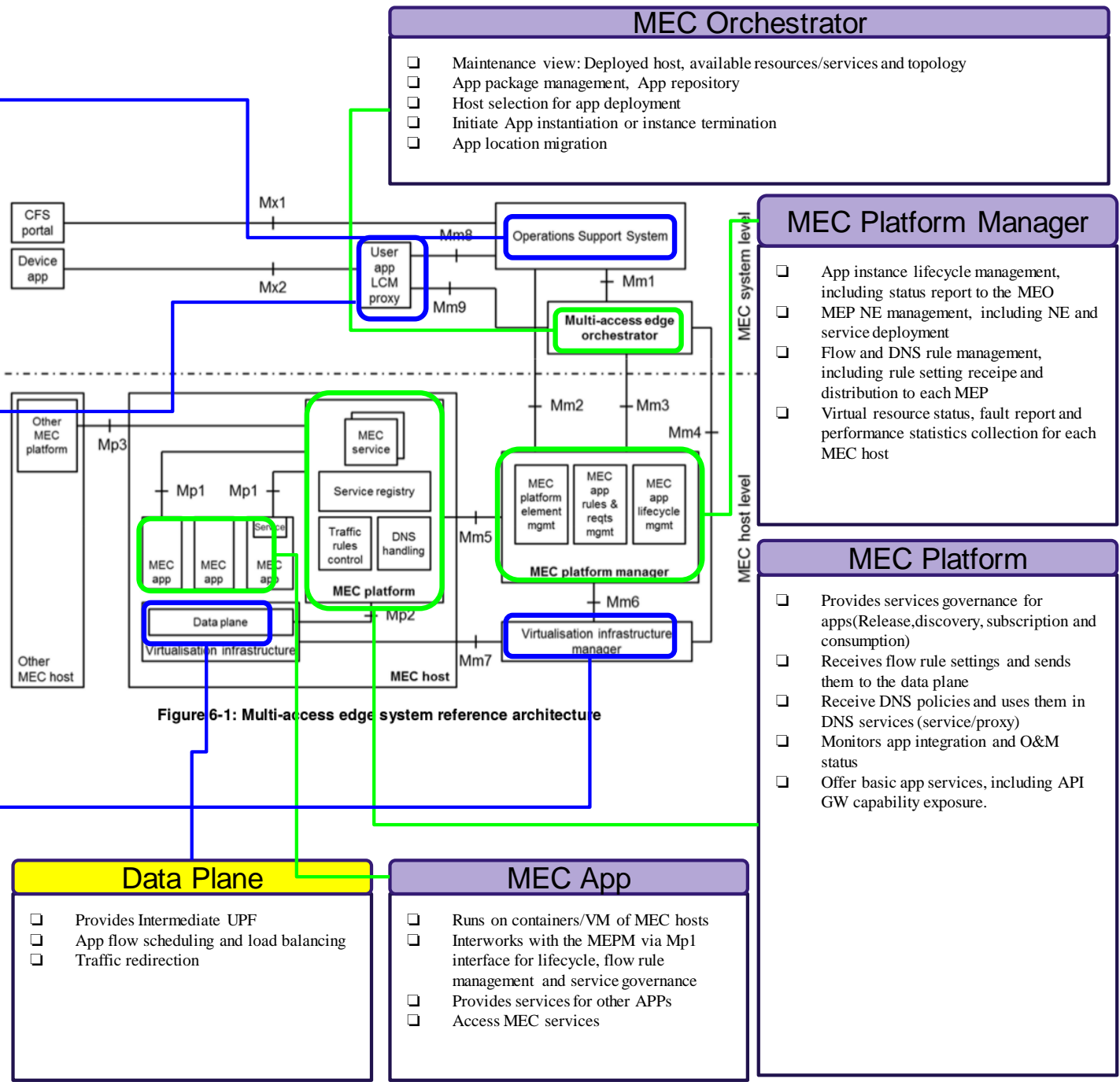


Figure 6-1: Multi-access edge system reference architecture

- ### MEC Orchestrator
- Maintenance view: Deployed host, available resources/services and topology
 - App package management, App repository
 - Host selection for app deployment
 - Initiate App instantiation or instance termination
 - App location migration

- ### MEC Platform Manager
- App instance lifecycle management, including status report to the MEO
 - MEP NE management, including NE and service deployment
 - Flow and DNS rule management, including rule setting receipt and distribution to each MEP
 - Virtual resource status, fault report and performance statistics collection for each MEC host

- ### MEC Platform
- Provides services governance for apps (Release, discovery, subscription and consumption)
 - Receives flow rule settings and sends them to the data plane
 - Receive DNS policies and uses them in DNS services (service/proxy)
 - Monitors app integration and O&M status
 - Offer basic app services, including API GW capability exposure.

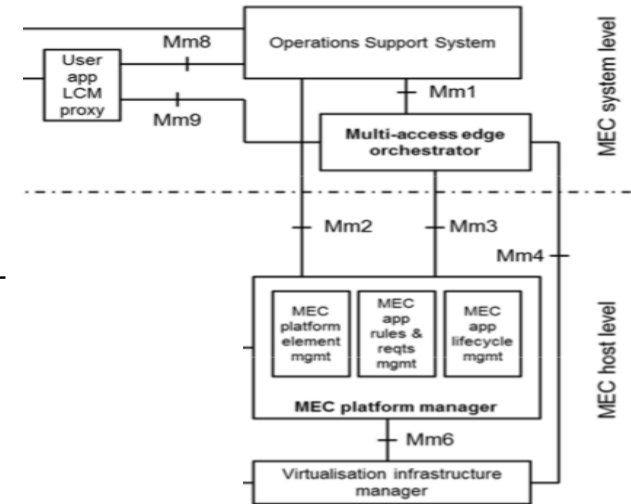
- ### Data Plane
- Provides Intermediate UPF
 - App flow scheduling and load balancing
 - Traffic redirection

- ### MEC App
- Runs on containers/VM of MEC hosts
 - Interworks with the MEPM via Mp1 interface for lifecycle, flow rule management and service governance
 - Provides services for other APPs
 - Access MEC services

MECM Requirement & Reference Points

The MECM is responsible for the following functions:

- Maintaining an overall view of the MEC system based on deployed MEC hosts, available resources, available MEC services, and topology;
- On-boarding of application packages, including checking the integrity and authenticity of the packages, validating application rules and requirements and if necessary adjusting them to comply with operator policies, keeping a record of on-boarded packages, and preparing the virtualization infrastructure manager(s) to handle the applications;
- Selecting appropriate MEC host(s) for application instantiation based on constraints, such as latency, available resources, and available services;
- Triggering application instantiation and termination;
- Triggering application relocation as needed when supported.



PS: MECM = MEAO + MEPM

Mm1 Reference Point

- App Package mgmt
 - On boarding
 - Search, delete, enable, disable
- App Life cycle mgmt
 - App Instantiation
 - Termination
 - Status change

Mm4 Reference Point

- manage virtualized resources of the MEC host, including keeping track of available resource capacity
- manage application images.

Mm3 Reference Point

- App Package mgmt
 - App package data requesting
 - App package change notice
 - App pkg on-boarding notice
 - App pkg retrieval
- App Life cycle mgmt (MEO->MEPM)
 - App instantiation and termination
 - App instance status change
 - Status request (MEO->MEPM)
 - App lifecycle change notice

MEC E2E Flow: Application Instantiation

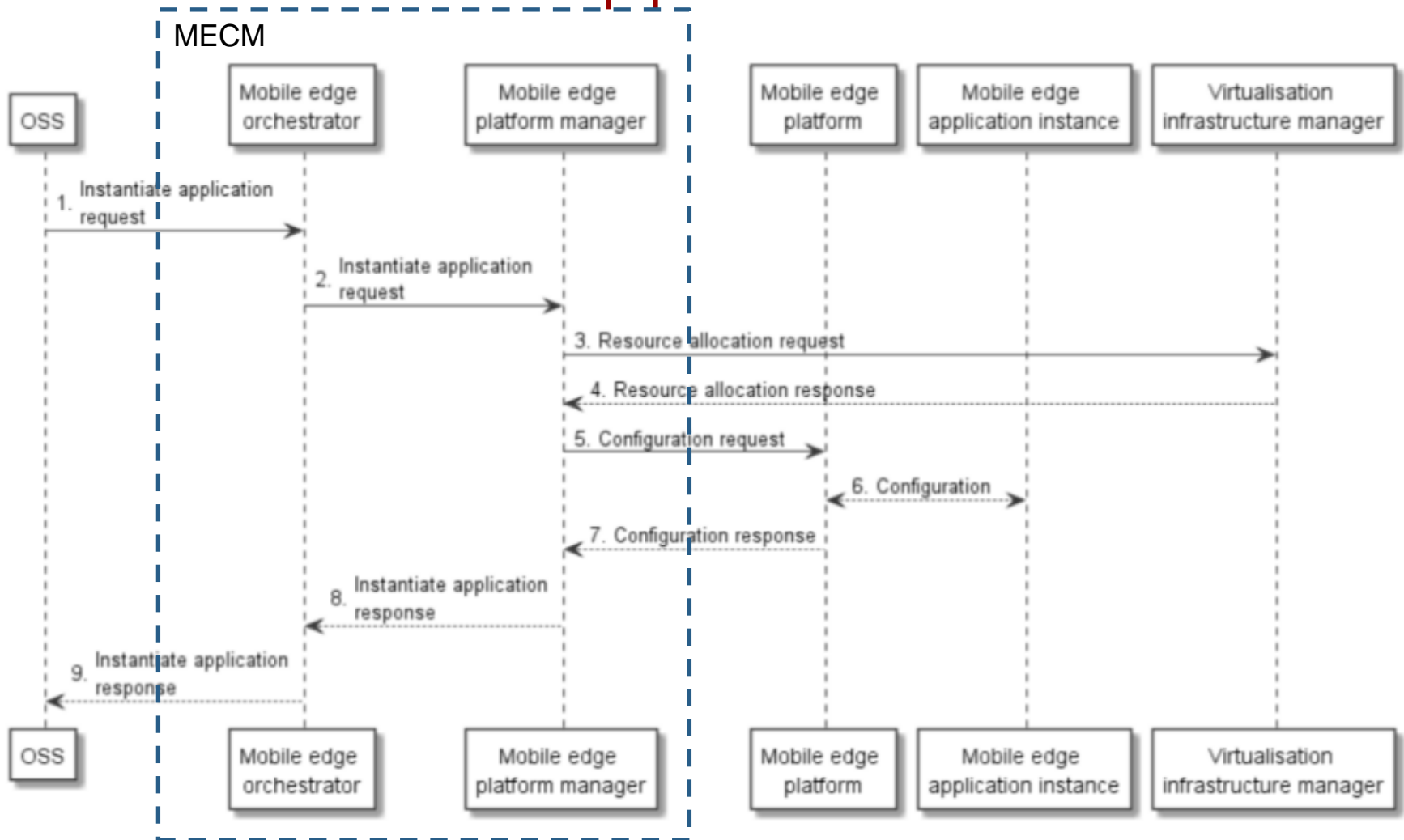
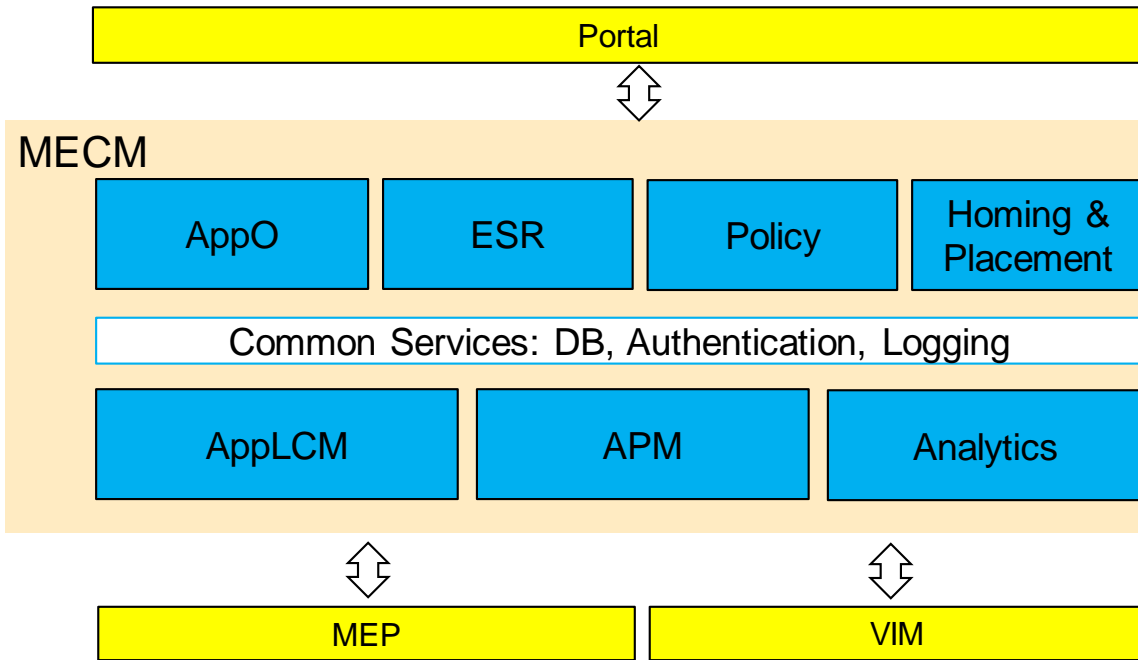


Figure 5.3-1: Application instantiation flow



Reference Implementation using ONAP

MECM Design



Analytics

- Responsible for handling Application Lifecycle Change Notification including Notification about change of application instance etc.

Homing & Placement

- Provides functionality for selecting appropriate MEC host(s) for application instantiation based on constraints, such as latency, available resources, and available services etc.

AppO

- Application orchestrator is the core module responsible for orchestrating life cycle maintenance operation of application.

ESR

- External system registry responsible for external system registrations like MEC Host, AppLCM, AppStore etc.

APPLCM

- Application life cycle manager handles life cycle of application including Instantiation, Termination, Status etc. based on the underlying cloud Infrastructure

APM

- Application package manager responsible for application package management including Onboarding, distributing package to edge etc.

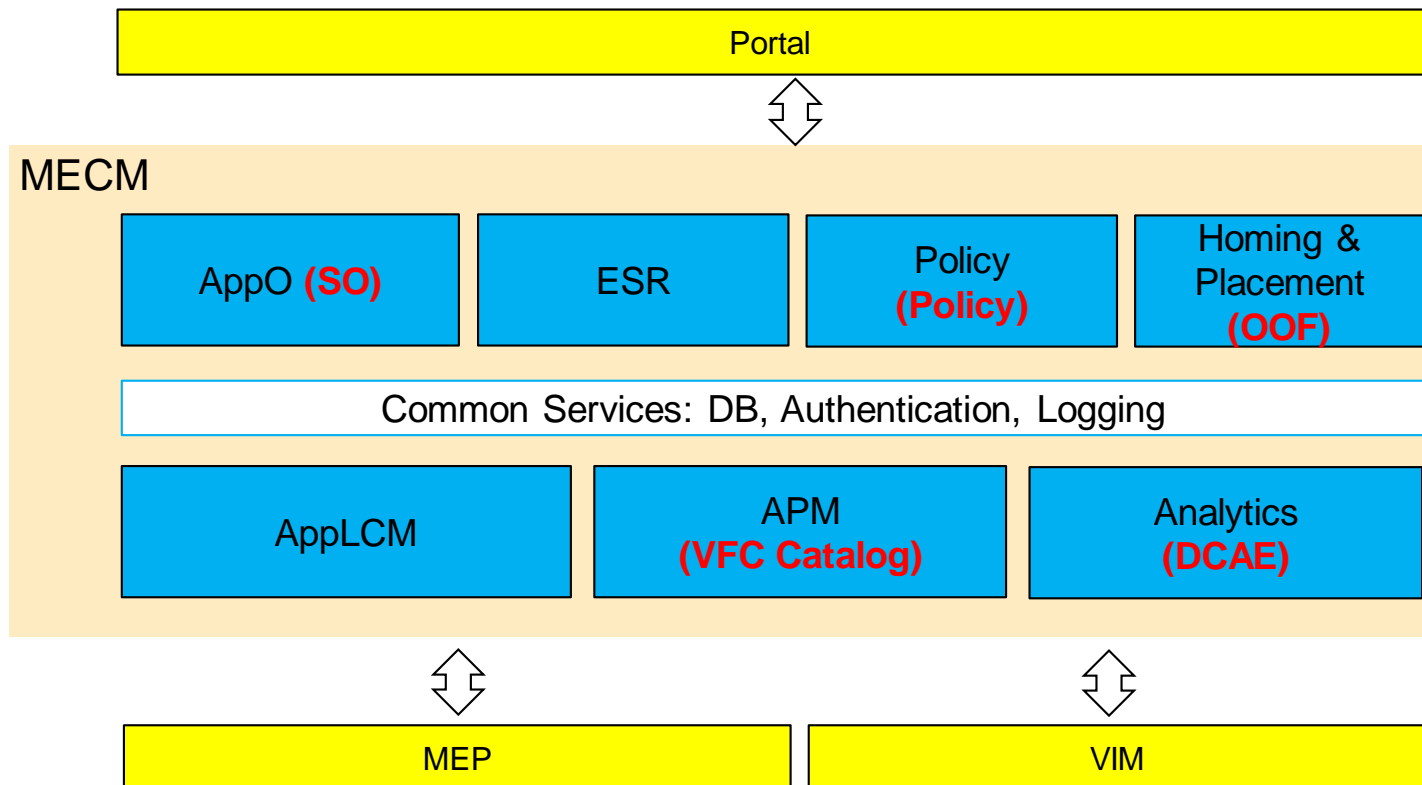
Policy

- Provide policy to take action based on analytics like application relocation.

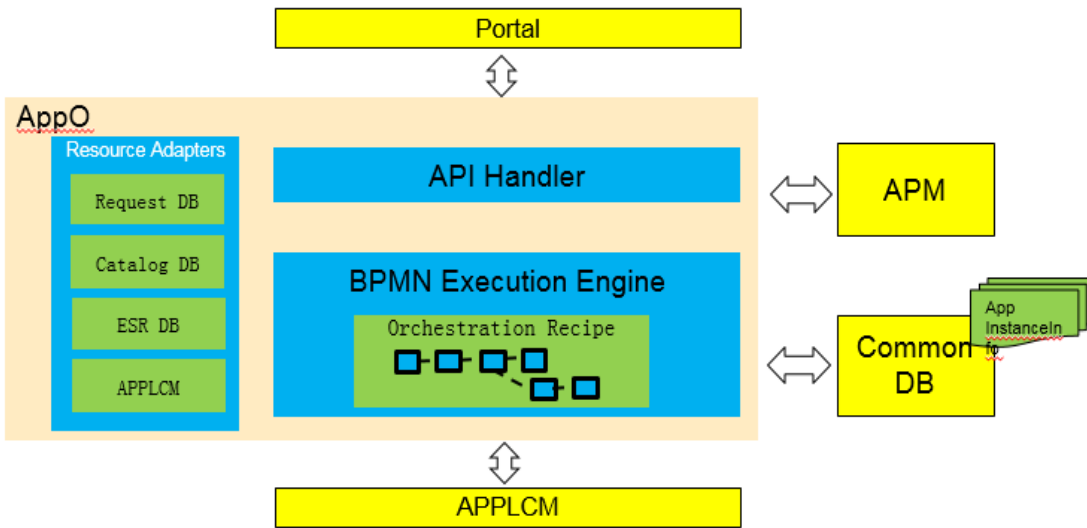
Common Services

- Provides common services like DB, Logging, Authentication for all the modules in MECM.

Smallest MEC Manager Realization using ONAP



AppO Design(Based on ONAP SO)



Application orchestrator aims at triggering application life cycle operations, maintain overall view of MEC system based on MEC host, select appropriate MEC host(s) for application life cycle based on constraints.

AppO Subcomponents

API Handler

- RESTFull interface to northbound clients.
- Handle Application life cycle related requests.

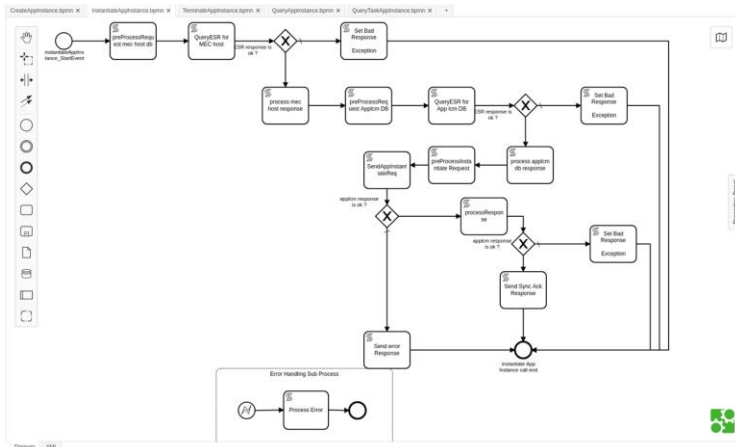
BPMN Execution Engine

- Expose RESTFull interface to recipe.
- Sequence orchestration steps to perform LCM actions on application.
- Maintain overall state of application LCM

Adapters

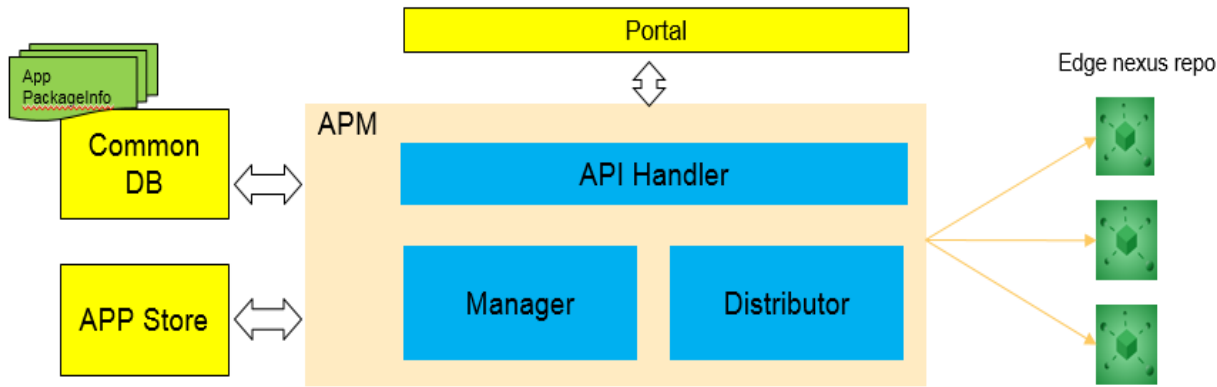
- Interface to other MECM components.

ONAP Service orchestrator is tailored to realize AppO orchestration functionality.



Recipe can be designed using camunda modeler as shown.

APM Design (Based on ONAP VFC Catalog)



Application Package Manager manages application download from appstore, Onboarding and maintain state of On boarded packages.

APM Subcomponents

API Handler

- RESTFull interface to northbound clients.
- Handle Application package management requests.

Manager

- Responsible for Application package download from appstore, Onboarding and maintain operational and usage status.

Distributor

- Responsible for distributing application package to destined edges.

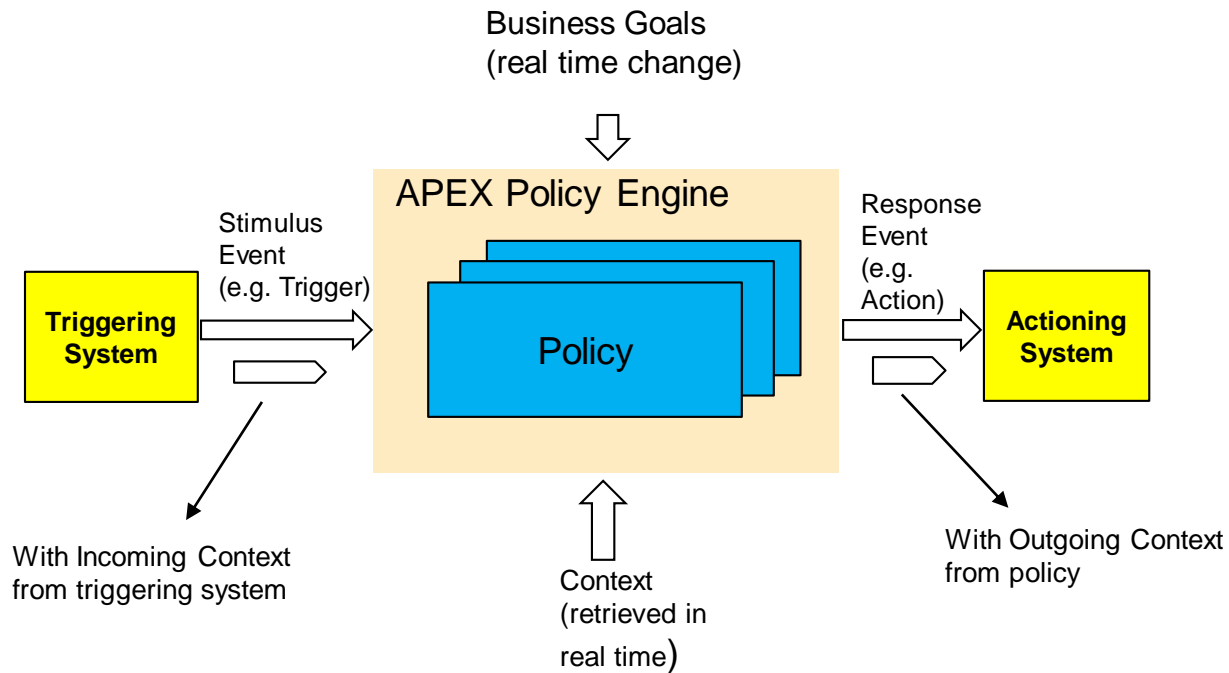
ONAP VFC catalog is tailored to realize APM functionality.

The screenshot shows the MEC Manager V0.2 interface. The breadcrumb navigation is 'MEC Manager / App Management / Package Distribution'. There is a search bar with 'Name' and 'Affinity' fields, and 'Reset' and 'Search' buttons. Below is a table with the following data:

Package Name	App Version	Provider	App Affinity	Modified Time	MEC Hosts	Status	Operation
face_recognition1.3	1.3	Huawei	GPU	2019-12-26 18:36:30	Host Ip 159.138.5.41	● Distributed	Delete Deploy

At the bottom right of the table, it says 'Total 1' and '10page' with navigation buttons.

Policy Design (Based on ONAP Policy)



- The Adaptive Policy Engine in APEX runs configured policies.
- These policies are triggered by incoming events.
- The logic of the policies executes and produces a response event.
- The Incoming Context on the incoming event and the Outgoing Context on the outgoing event are simply the fields and attributes of the event.
- You design the policies that APEX executes, trigger and action events that your policies accept and produce

Thanks!

Any questions?