

# R5 Federated MEC Platform BP Blog

- [Introduction](#)
  - [What is MEC and the challenges?](#)
- [Blueprint Overview: Federated Multi-Access Edge Cloud Platform](#)
- [Functional Diagram: Federated Multi-Access Edge Cloud Platform](#)
- [Key Enabling Architectural Components](#)
  - [Federation Scheduler \(Included in Release-5\)](#)
  - [EdgeMesh \(Included in Release-5\)](#)
  - [Service Discovery \(Not included in Release-5\)](#)
  - [Mobility Management \(Not included in Release-5\)](#)
  - [Multi-Access Gateway \(Not included in Release-5\)](#)
  - [AutoScaling \(Not included in Release-5\)](#)
  - [Service Catalog \(Not included in Release-5\)](#)
- [Detail Flow of various Architectural Components](#)
- [What is included in Release-5](#)
- [Acknowledgements](#)

## Introduction

This blog focuses on providing an overview of the "[Federated Multi-Access Edge Cloud Platform](#)" blueprint as part of the Akraino Public Cloud Edge Interface (PCEI) blueprint family. This blog specifically provides an overview of the key features and implemented components as part of Akraino Release-5. Key idea for this blueprint is to have a federated Multi-Access Edge Cloud (MEC) Platform that showcases the inter-play between telco side and cloud /edge side.

Prior to discussing the specifics of this blueprint implementation, the following subsection provides a brief description of what Multi-Access Edge Cloud (MEC) is and how it is ushering in as an enabler for emerging 5G/AI based applications landscape.

## What is MEC and the challenges?

MEC is a network architecture concept that enables cloud computing and IT capabilities to run at the edge of the network. Applications or services running on the edge nodes – which are closer to end users – instead of on the cloud, can enjoy the benefits of lower latency and enhanced end-user experience. MEC essentially moves the intelligence and service control from centralized data centers to the edge of the network – closer to the users. Instead of backhauling all the data to a central site for processing, it can be analyzed, processed, stored locally and shared upstream when needed. MEC solutions are closely "integrated" with access network(s). Such environments often include WiFi, mobile access protocols such as 4G-LTE/5G etc.

MEC opens up plethora of potential vertical and horizontal use cases, such as Autonomous Vehicle (AV), Augmented Reality (AR) and Virtual Reality (VR), Gaming, and Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) enabled applications like autonomous navigation, remote monitoring by using Natural Language Processing (NLP) or facial recognition, video analysis, and more. These emerging 5G/AI based applications landscape typically exhibit characteristics such as the following:

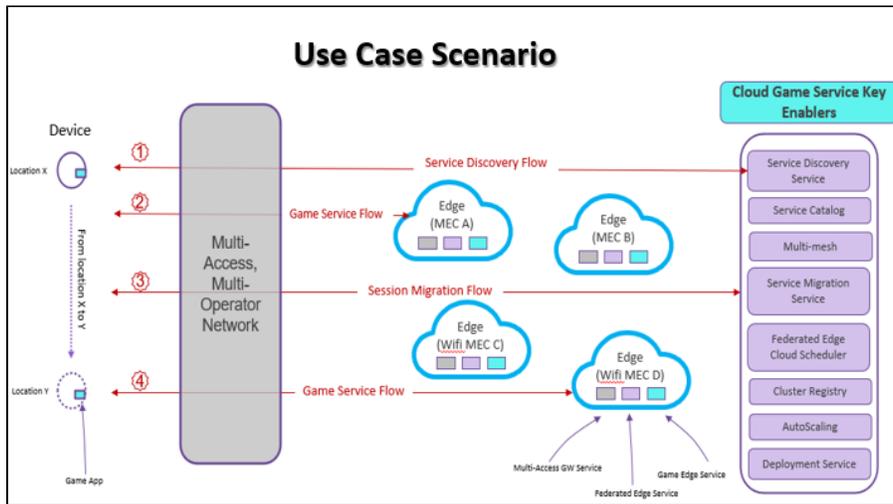
- Low Latency requirements
- Mobility
- Location Awareness
- Privacy/Security etc.

All of these characteristics pose unique challenges while developing emerging 5G/AI applications at the network edge. Supporting such an extensive feature-set at the required flexibility, dynamicity, performance, and efficiency requires careful and expensive engineering effort and needs adoption of new ways of architecting the enabling technology landscape.

To this regards, our proposed "[Federated Multi-Access Edge Cloud Platform](#)" blueprint enables desired abstractions in order to address these challenges and, as a result, ushers in an application development environment that enables support for ease of development and deployment of these emerging applications landscape. Subsequent sections delve deep into the proposed "[Federated Multi-Access Edge Cloud Platform](#)" blueprint details.

## Blueprint Overview: Federated Multi-Access Edge Cloud Platform

The purpose of the "[Federated Multi-Access Edge Cloud Platform](#)" blueprint is an end-to-end technology solution for mobile game deployed across multiple heterogeneous edge nodes using various network access protocols such as mobile and WiFi and others. This blueprint demonstrates how an application leverages a distributed and multi access network edge environment in order to get all the benefits of edge computing.



The diagram above highlights the use case scenario. On the left hand side is the device – as can be seen that the device is moving from location x to y.

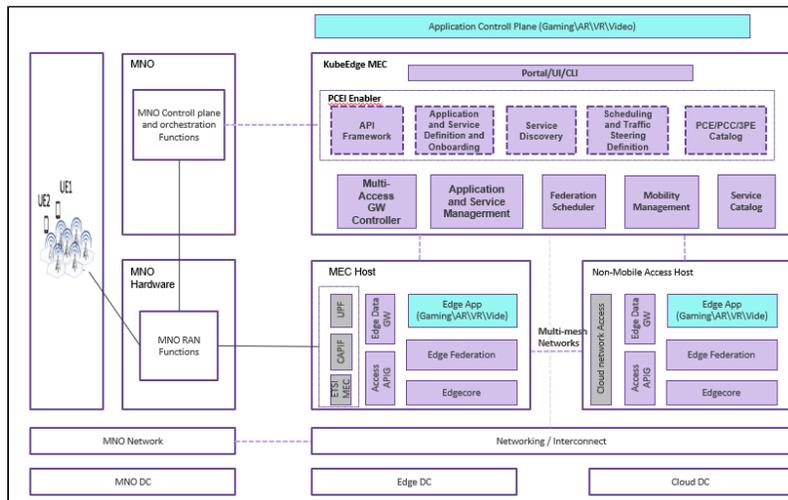
The whole use case goes through 4 distinct steps. The first step is the service discovery flow. And then the game service flow follows. And once the device actually moves, it would trigger additional session migration flow. This also includes subsequent service discovery to go along with this session migration. Finally, step number four is once this migration happens, the UE will go to the new edge node.

In order to support all this, platform provides two key abstractions:

- **Multi-Access/Mobile Operator Network Abstraction:** Multi-access network means a mobile, Wi Fi or whatever it takes. Multi-operator means even for the same 4G/5G there could be different operators (Verizon, AT&T etc.). There are various MEC edge nodes, they could be the WiFi based edge node and they can be from different operators.
- **Cloud-Side Abstraction:** Cloud side abstraction includes key architectural components to be described in the subsequent sections.

## Functional Diagram: Federated Multi-Access Edge Cloud Platform

The key component is this federated multi-access edge platform. The platform sits between applications and underlying heterogeneous edge infrastructure and also abstracts the multi-access interface and exposes application developer friendly APIs. This blueprint leverages upstream project KubeEdge as baseline platform – this includes the enhanced federation function (Karmada).

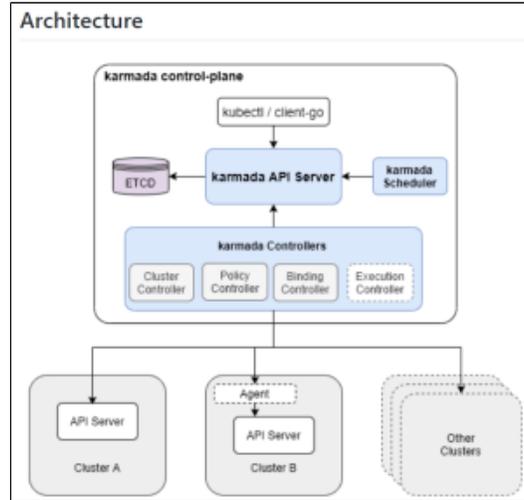


Telco/GSMA side complexities (5GC/NEF etc.) need to be thought through and designed appropriately in order to realize extremely low latencies (10 ms) requirements desired by typical MEC use cases. For the multi access, we may initially use a simulated mobile access environment to mimic a real time device access protocol conditions as part of the initial release/s.

## Key Enabling Architectural Components

### Federation Scheduler (Included in Release-5)

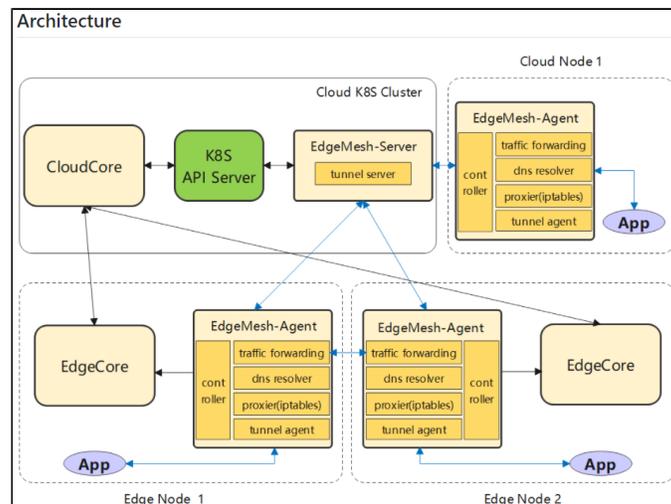
As a “Global Scheduler”, responsible for application QoS oriented global scheduling in accordance to the placement policies. Essentially, it refers to a decision-making capability that can decide how workloads should be spread across different clusters similar to how a human operator would. It maintains the resource utilization information for all the MEC edge cloud sites. Cloud federation functionality in our blueprint is enabled using open source Karmada project. The following is an architecture diagram for Karmada.



Karmada (Kubernetes® Armada) is a Kubernetes® management system that enables cloud-native applications to run across multiple Kubernetes® clusters and clouds with no changes to the underlying applications. By using Kubernetes®-native APIs and providing advanced scheduling capabilities, Karmada truly enables multi-cloud Kubernetes® environment. It aims to provide turnkey automation for multi-cluster application management in multi-cloud and hybrid cloud scenarios with key features such as centralized multi-cloud management, high availability, failure recovery, and traffic scheduling. More details related to Karmada project can be found [here](#).

## EdgeMesh (Included in Release-5)

EdgeMesh provides support for service mesh capabilities for the edge clouds in support of microservice communication cross cloud and edges. EdgeMesh provides a simple network solution for the inter-communications between services at edge scenarios (east-west communication).



The network topology for edge cloud computing scenario is quite complex. Various Edge nodes are often not interconnected and the direct inter-communication of traffic between applications on these edge nodes is highly desirable requirement for businesses. EdgeMesh addresses these challenges by shielding the complex network topology at the edge applications scenario. More details related to EdgeMesh project can be found [here](#).

## Service Discovery (Not included in Release-5)

**Service Discovery** retrieves the endpoint address of the edge cloud service instance depending on the UE location, network conditions, signal strength, delay, App QoS requirements etc.

## Mobility Management (Not included in Release-5)

Cloud Core side mobility service subscribes to UE location tracking events or resource rebalancing scenario. Upon UE mobility or resource rebalancing scenario, mobility service uses Cloud core side Service Discovery service interface to retrieve the address of new appropriate location-aware edge node. Cloud Core side mobility service subsequently initiates UE application state migration process between edge nodes. Simple CRIU container migration strategy may not be enough, it is much more complex than typical VM migration.

## Multi-Access Gateway (Not included in Release-5)

Multi access gateway controller manages Edge Data Gateway and Access APIG of edge nodes. Edge data gateway connects with edge gateway (UPF) of 5G network system, and routes traffic to containers on edge nodes. Access APIG connects with the management plane of 5G network system (such as CAPIF), and pulls QoS, RNIS, location and other capabilities into the edge platform.

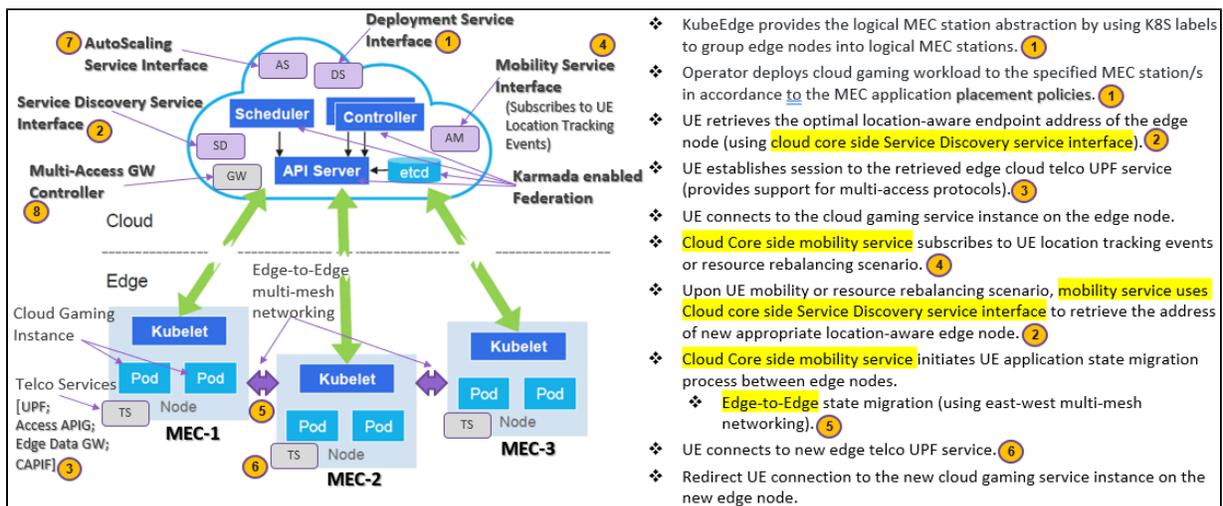
## AutoScaling (Not included in Release-5)

Autoscaling provides capability to automatically scale the number of Pods (workloads) based on observed CPU utilization (or on some other application-provided metrics). Autoscaler also provides vertical Pod autoscaling capability by adjusting a container's "CPU limits" and "memory limits" in accordance to the autoscaling policies.

## Service Catalog (Not included in Release-5)

**Service Catalog** provides a way to list, provision, and bind with services without needing detailed knowledge about how those services are created or managed.

## Detail Flow of various Architectural Components



## What is included in Release-5

As mentioned earlier that the purpose of this blueprint is an end-to-end technology solution for mobile game deployed across multiple heterogeneous edge nodes using various network access protocols such as mobile and WiFi and others. This blueprint demonstrates how an application leverages a distributed and multi access network edge environment for realizing all the benefits of edge computing.

This is the very first release of this new blueprint as part of the Akraino PCEI family. Current focus for this release is to enable only the following two key architectural components:

1. Open source Karmada based Cloud Federation
2. EdgeMesh functionality

This blueprint will evolve as we incorporate remaining architectural components as part of the subsequent Akraino releases. More information on this blueprint can be found [here](#).

## Acknowledgements

**Project Technical Lead:** Deepak Vij, KubeEdge MEC-SIG member - Principal Cloud Technology Strategist at Futurewei Cloud Lab.

**Contributors:**

- Peng Du, Futurewei Cloud Lab.

- Hao Xu, Futurewei Cloud Lab.
- Qi Fei, KubeEdge MEC-SIG member - Huawei Technologies Co., Ltd.
- Xue Bai, KubeEdge MEC-SIG member - Huawei Technologies Co., Ltd.
- Gao Chen, KubeEdge MEC-SIG member - China Unicom Research Institute
- Jiawei Zhang, KubeEdge MEC-SIG member - Shanghai Jiao Tong University
- Ruolin Xing, KubeEdge MEC-SIG member - State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications
- Shangguang Wang, KubeEdge MEC-SIG member - State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications
- Ao Zhou, KubeEdge MEC-SIG member - State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications
- Jiahong Ning, KubeEdge MEC-SIG member - Southeastern University
- Tina Tsou, Arm