LF NETWORKING
Developer & Testing Forum

Nephio: Birds of Feather Session: Leveraging EMCO Code to Accelerate Nephio

Amar Kapadia (Aarna Networks), Piotr Matysiak (Orange),
Sandeep Sharma (Aarna), Sundar Nadathur (Intel)

February 15, 2023

https://lfnetworking.org

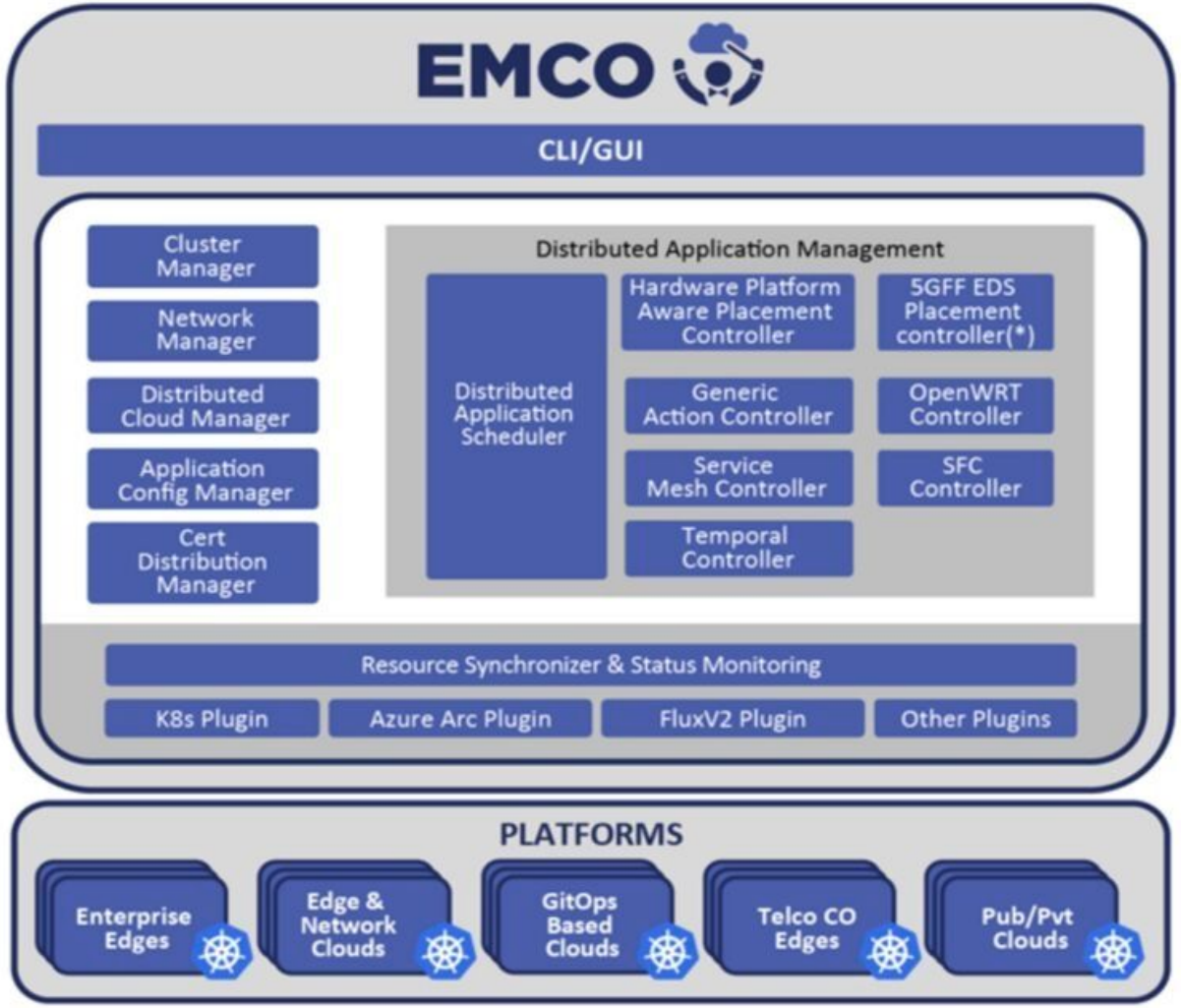
Anti-Trust Policy Notice

LFN NETWORKING
LFN Developer & Testing Forum

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrustpolicy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

Refresher on EMCO



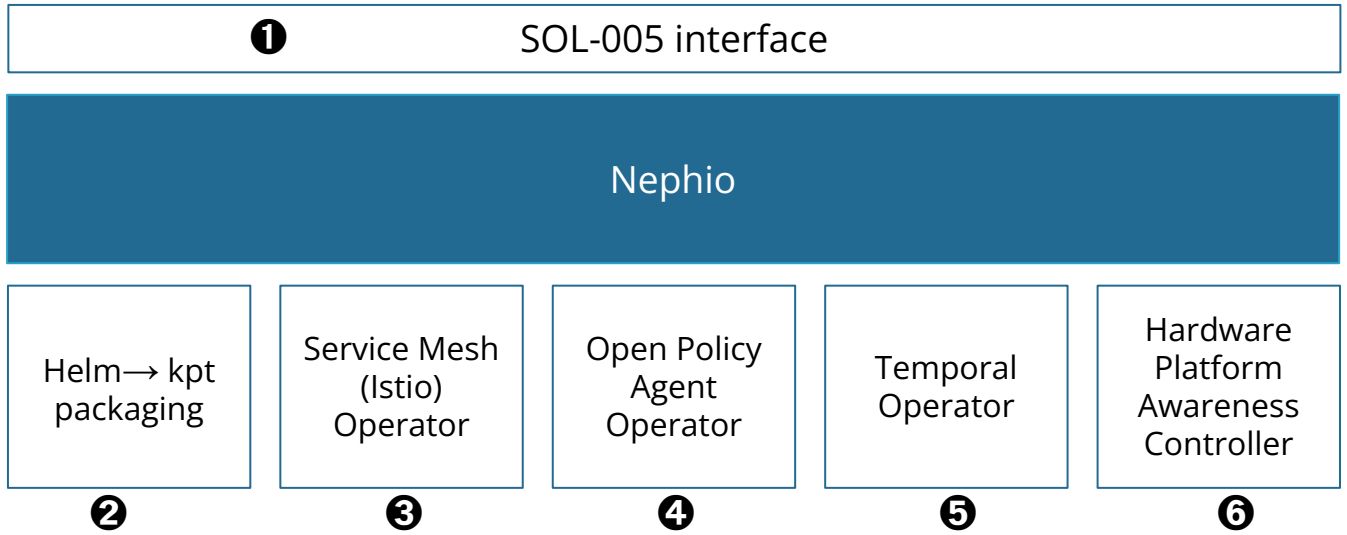
"The Edge Multi-Cluster Orchestrator (EMCO) is a software framework for intent-based deployment of cloud-native applications to a set of Kubernetes clusters, spanning enterprise data centers, multiple cloud service providers and numerous edge locations. It is architected to be flexible, modular and highly scalable. It is aimed at various verticals, including telecommunication service providers."

Background

- EMCO (Edge Multi Cluster Orchestrator) is an LFN project that is working on orchestration of xNFs & CNAs (cloud native application) across a large number of distributed K8s clusters
- However, with Intel pulling out, EMCO is now in maintenance mode
- There is a lot of quality code that may be useful in accelerating Nephio
- We would like to discuss this topic in today's session to gauge interest

Core Ideas

- Leverage EMCO code to build Nephio Controllers
- Leverage SOL-005 code (to be upstreamed by Aarna)
- Leverage EMCO Helm parsing code for Helm→kpt conversion



1. SOL-005

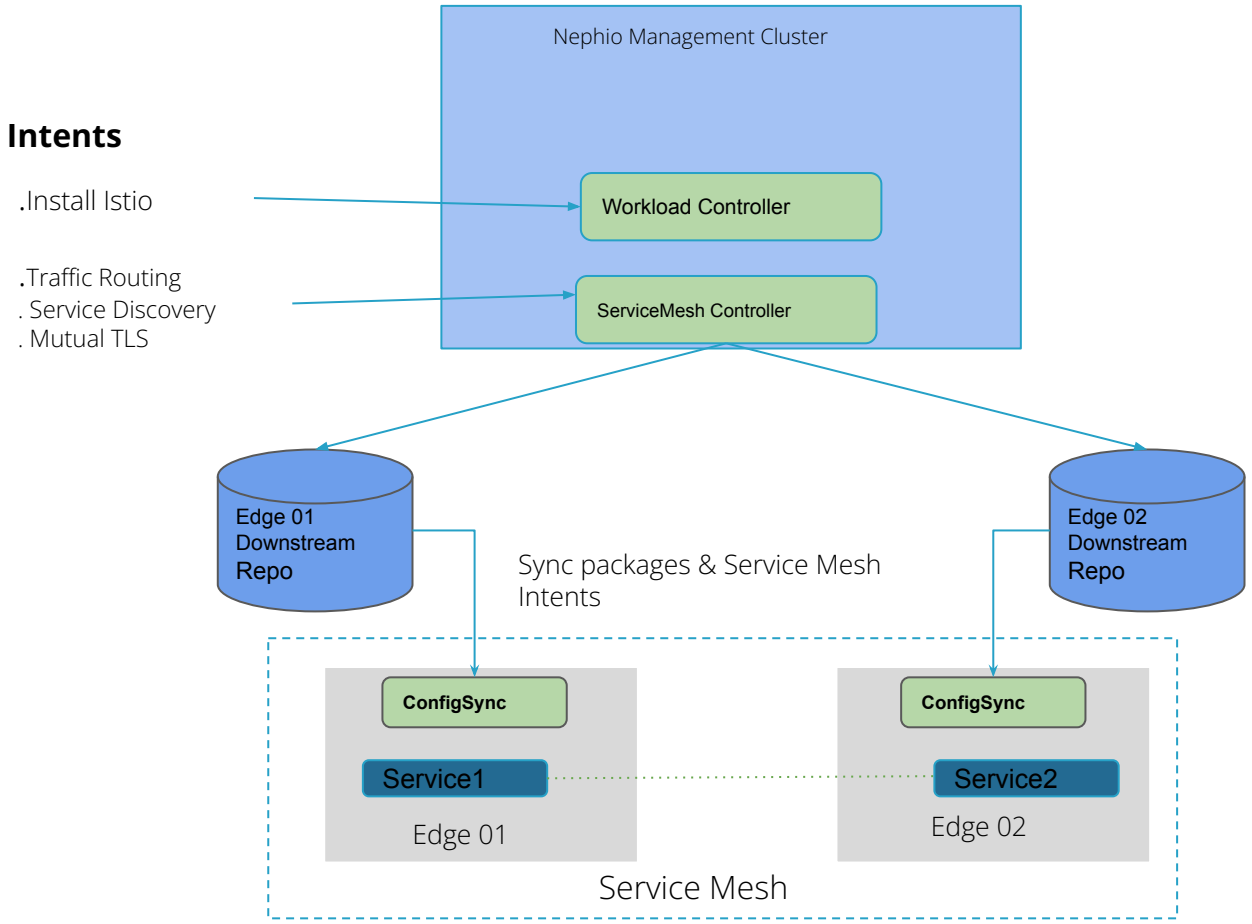
- This component interfaces to northbound service orchestrators using ETSI SOL-005 APIs
- The southbound could create Nephio declarative intent in kpt packages
- CSAR packaging with Helm charts embedded could be supported (not sure about full TOSCA support) – See Helm→kpt idea

2. Helm→kpt Packaging

Using Helm parsing code from EMCO to create Helm→kpt

3. Istio Controller

- Install Istio in K8s clusters: This is no different from installing any other workload
- We propose a new controller which will reconcile high level service mesh intents
  - L7 traffic routing Intents
  - Cross Cluster Service Discovery Intents: Enable users to specify service discovery Intents, the service mesh controller will reconcile them and connect services across clusters in the mesh
  - Mutual TLS intents

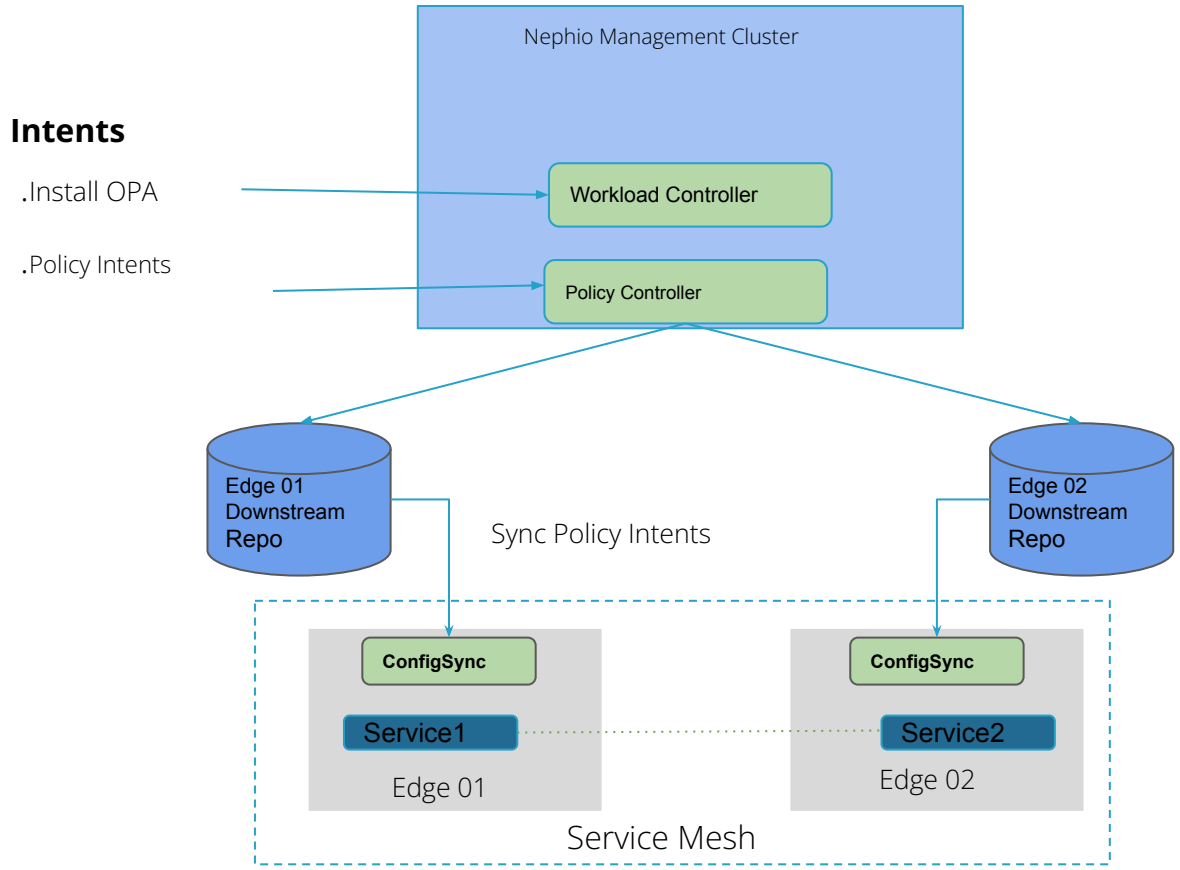


4. Open Policy Agent Controller

We propose a Policy engine to be deployed in Workload clusters in order to enforce policies across the stack. The engine could be OPA.

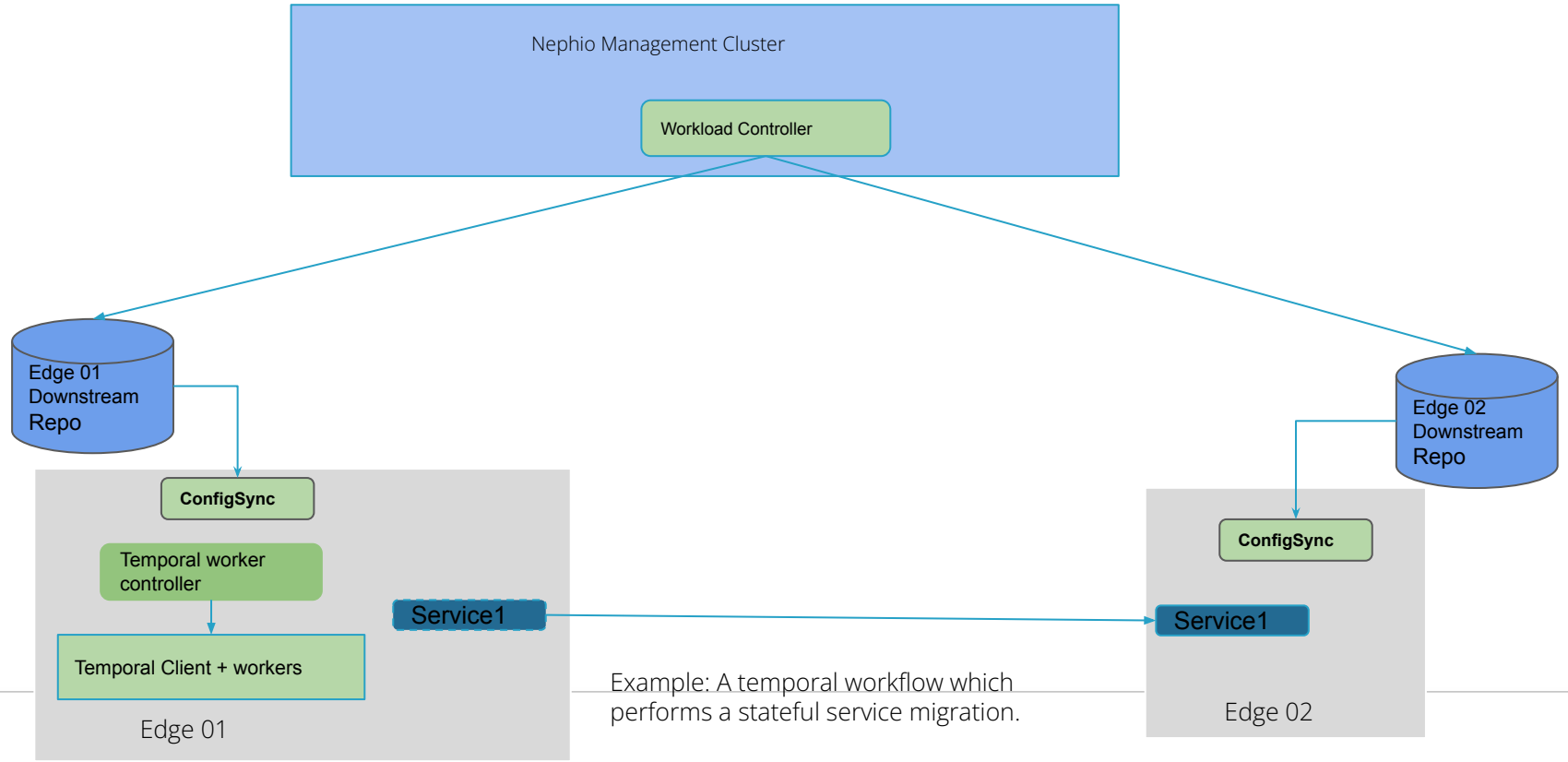
A controller will reconcile the Policy related Intents; following are few examples:

- OPA + Istio: Service authorization policies with CUSTOM action of ISTIO and OPA as external Authorizer
- OPA + Workload Controller: Workload controller can delegate the task of finding the best cluster for deployment to OPA; user can dynamically deploy new policies
- OPA as general policy evaluation for service assurance related use cases



5. Temporal Operator

The LCM of services deployed in the workload clusters might require a sequence of complex actions to be performed across hardware and software. These complex actions can be deployed as workflows and executed by a workflow engine. We propose automating this process via workflow operator in Nephio. The workflow engine could be Temporal. See also **Edge Relocation**.

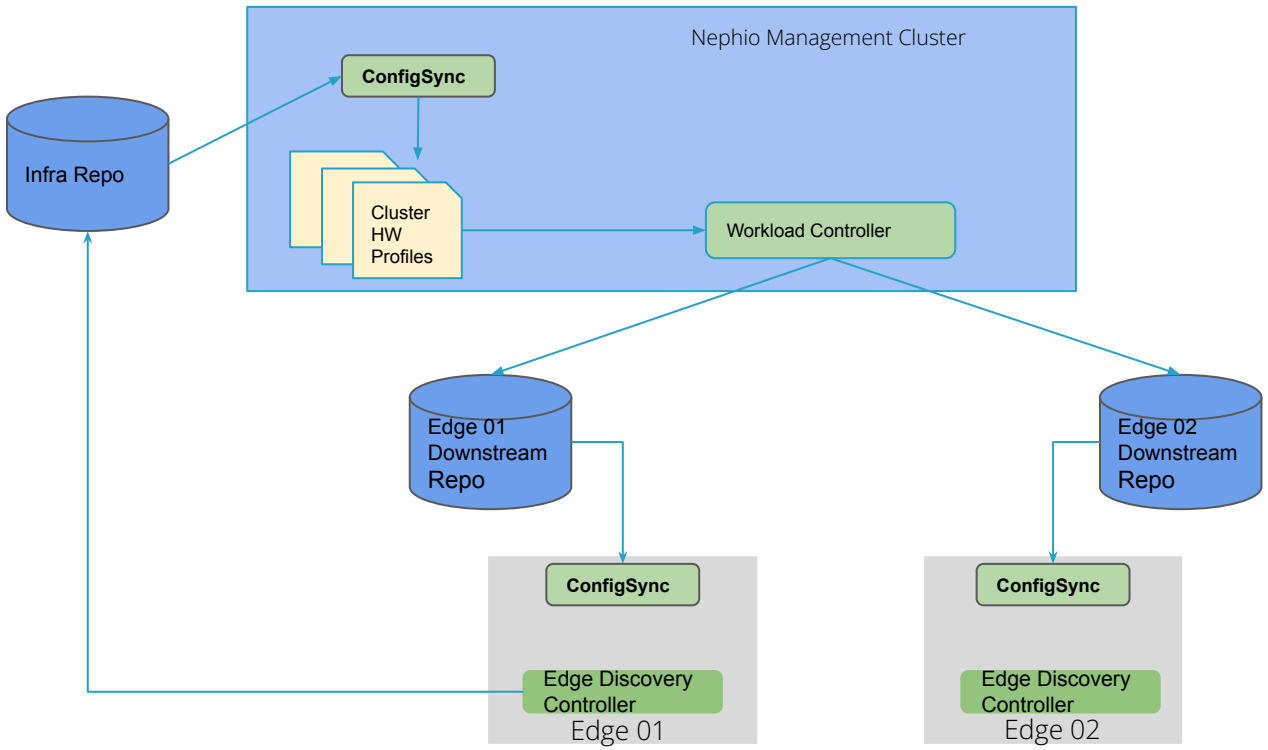


Example: A temporal workflow which performs a stateful service migration.

6. Hardware Placement Awareness Controller

This solution builds the workload cluster hardware capability knowledge base in the Nephio mgmt cluster. The workload cluster will consult this information either directly or via the OPA controller in order to resolve the workload Intents.

1. A new controller called Edge discovery controller will be deployed in workload clusters. This cluster along with the Node feature discovery agent will discover hardware capabilities and update the cluster profiles in the Infra Repo. Cluster profiles will be K8s CRs.
2. Config sync in Nephio mgmt cluster will keep reconciling the information and update the cluster profiles.
3. OPA or the workload controller will look into the cluster profile CRs to address the workload Intent. For example:
   a. Intent: Deploy MEC app on a cluster with DPDK.



LFN NETWORKING

LFN Developer & Testing Forum

Edge Relocation

Edge Relocation: motivations

Problem statement according to ETSI standards

The user (UE) is consuming a service, while moving out of the coverage area of Source MEC Host (Cluster A). Later it enters the coverage area of Target MEC Host (Cluster B) and expects to continue consuming the same service. This requires a relocation of a service instance from cluster A to cluster B.

Motivation for Edge Relocation “user perspective”

- Services leveraging Edge support in 5G and future 6G (e.g., XR, cloud-gaming, V2X etc.), generally require ***stringent QoS*** with ***very low latency*** and ***high availability***;
- The above need to be assured in dynamic environment, where **users are constantly moving**;
- In this context, application relocation is key to enrich Edge Systems with smart automated application placement while achieving an enhanced quality of experience.

Motivation for Edge Relocation “network perspective”

- Network functions may need to be dynamically moved to enhance performances and maintain Service Level Agreement
- It offers higher degree of network function placement (e.g., Network Function Chaining)
- Relieving the network by local traffic handling;
- Possibility to use smart techniques, to load-balance the workloads between local Edge Clusters.

Edge Relocation: requirements

Service continuity assurance

- Workload relocation between K8s clusters should assure zero or near-zero service downtime;
- The new instance of the App must be declared to be 'ready' before we can steer the traffic to the new App instance;

Smart destination selection

- If there are several candidates for the target MEC cluster, the final choice should be made by MEC Orchestrator;
- It requires constant observability over Kubernetes cluster resources, in order to load-balance the resource consumption;

Constant (central) reconciliation whether to relocate (when, where)

- Many different triggers: cluster disaster, cluster overload, user movement

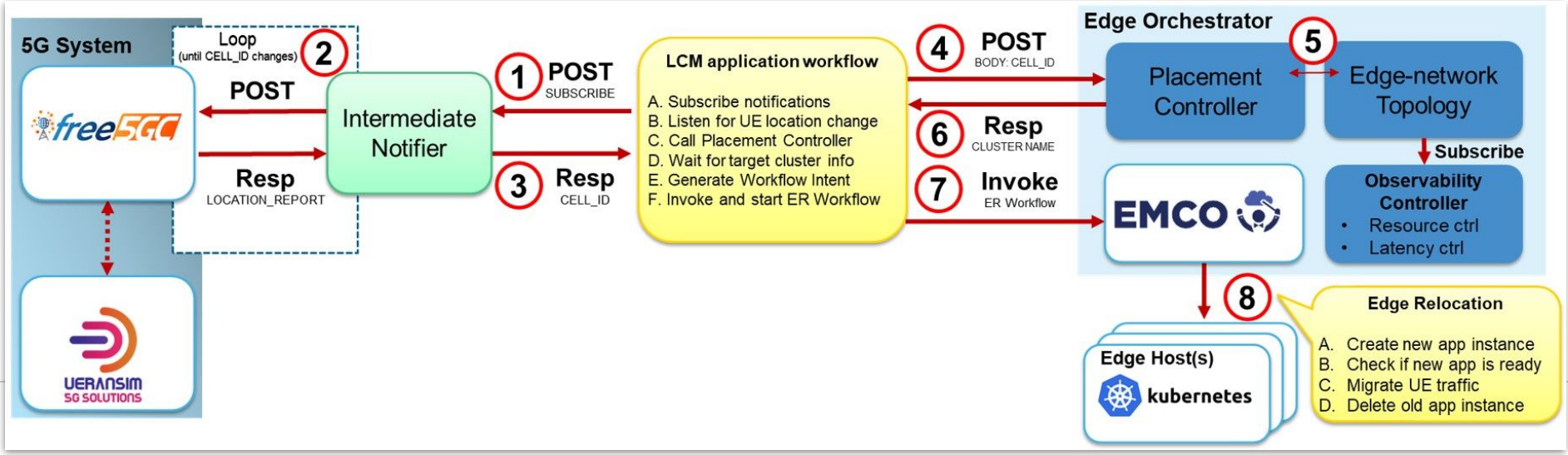
Edge Relocation: what is/could be done?

We leveraged **EMCO**, and its integration with **Temporal**, to build Smart Placement control loop on top of it.

What could be done in Nephio: To build K8s-native Smart Placement controllers / Smart Relocation procedures;

Possible **enablers**:

- To introduce *"Cluster Topology"* concept build-in into Nephio;
- To provide out-of-the-box infrastructure monitoring (e.g. integration with Prometheus & central TSDB like Mimir/Thanos/Cortex), to collect and utilize metrics for smart decision making;
- To feed with smart placement policies



LF NETWORKING

LFN Developer & Testing Forum

Discussion