

OLF NETWORKING

LFN Developer & Testing Forum

PCEI Edge to Cloud connectivity and Application deployment

Presented by:

Sriram Rupanagunta, Aarna Networks

Oleg Berzin, Fellow Technology and Architecture, Equinix

Brandon Wick, Aarna Networks

Vivekanandan Muthukrishnan, Aarna Networks

Contribution by:

Kavitha Papanna, Member Technical Staff, Aarna Networks

Vikas Kumar, Member Technical Staff, Aarna Networks

Premkumar Subramaniyan, Member Technical Staff, Aarna Networks

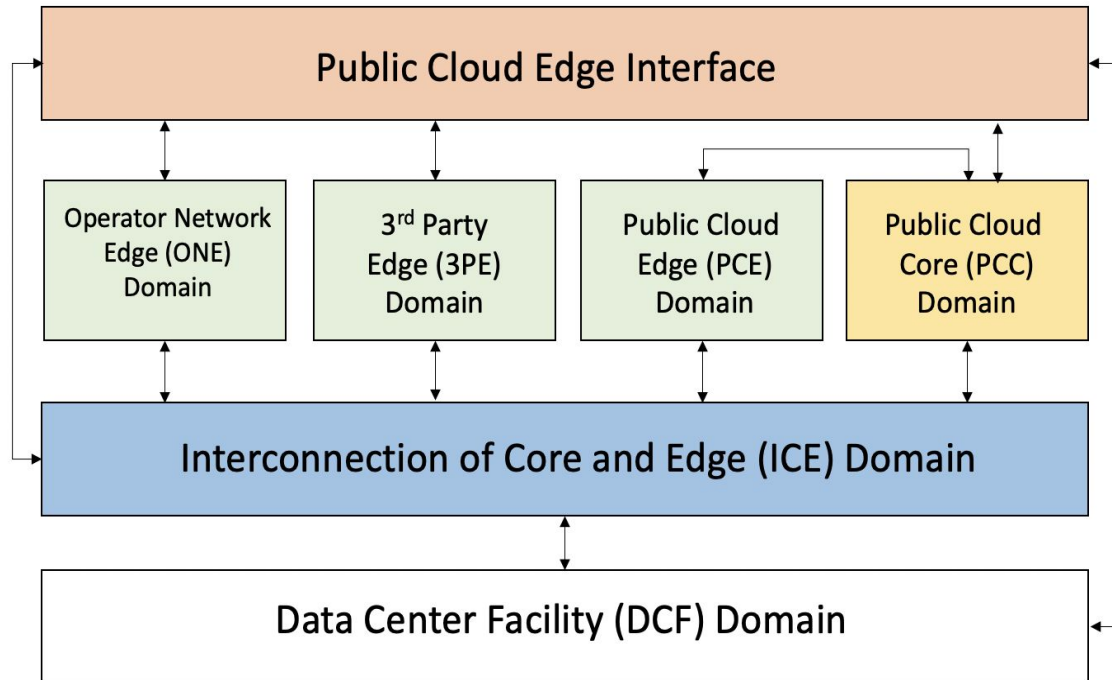
Sai Lakshmi Cheedella, Member Technical Staff, Aarna Networks

- Public Cloud Edge Interface (**PCEI**) Blueprint Introduction
- Terraform based orchestration
- EMCO, ONAP CDS, Camunda & Terraform Integration Architecture
- Orchestration Workflow Sequence Diagram
- Camunda workflow state machine
- Demo - Video

Motivation for LFN PCEI Akraino Project

- ❑ **Major industry and market trends**
 - ❑ Public Cloud-Driven Edge Computing
 - ❑ Edge computing infrastructure & resources are increasingly provided by public clouds (e.g., AWS Outposts)
- ❑ **Hybrid infrastructure**
 - ❑ Most practical deployments of edge infrastructure and applications are hybrid in nature, where an application deployed at the edge needs services residing in the core cloud to function (coupled model)
- ❑ **Need for Multi-Domain Interworking**
 - ❑ Individual service domains (e.g., edge, cloud, network fabric) present their own APIs and/or other provisioning methods (e.g., CLI), thus making end-to-end deployment challenging both in complexity and in time.
 - ❑ A Multi-domain orchestration solution is required to handle edge, cloud and interconnection in a uniform and consistent manner.
- ❑ **Need for Interconnection and Federation**
 - ❑ Need for efficient and performant interconnection and resource distribution between edge and cloud as well as between distributed edges proximal to end users.
- ❑ **Need for Bare Metal orchestration**
 - ❑ Many orchestration solutions assume that the bare metal compute/storage hardware and basic operating system resources are available for the deployment of virtualization and application/services layers.
- ❑ **Need for developer-centric capabilities**
 - ❑ End-to-end application deployment, configuration and interconnection
 - ❑ Integration with CI/CD environments

Public Cloud Edge Interface – Akraino Blueprint



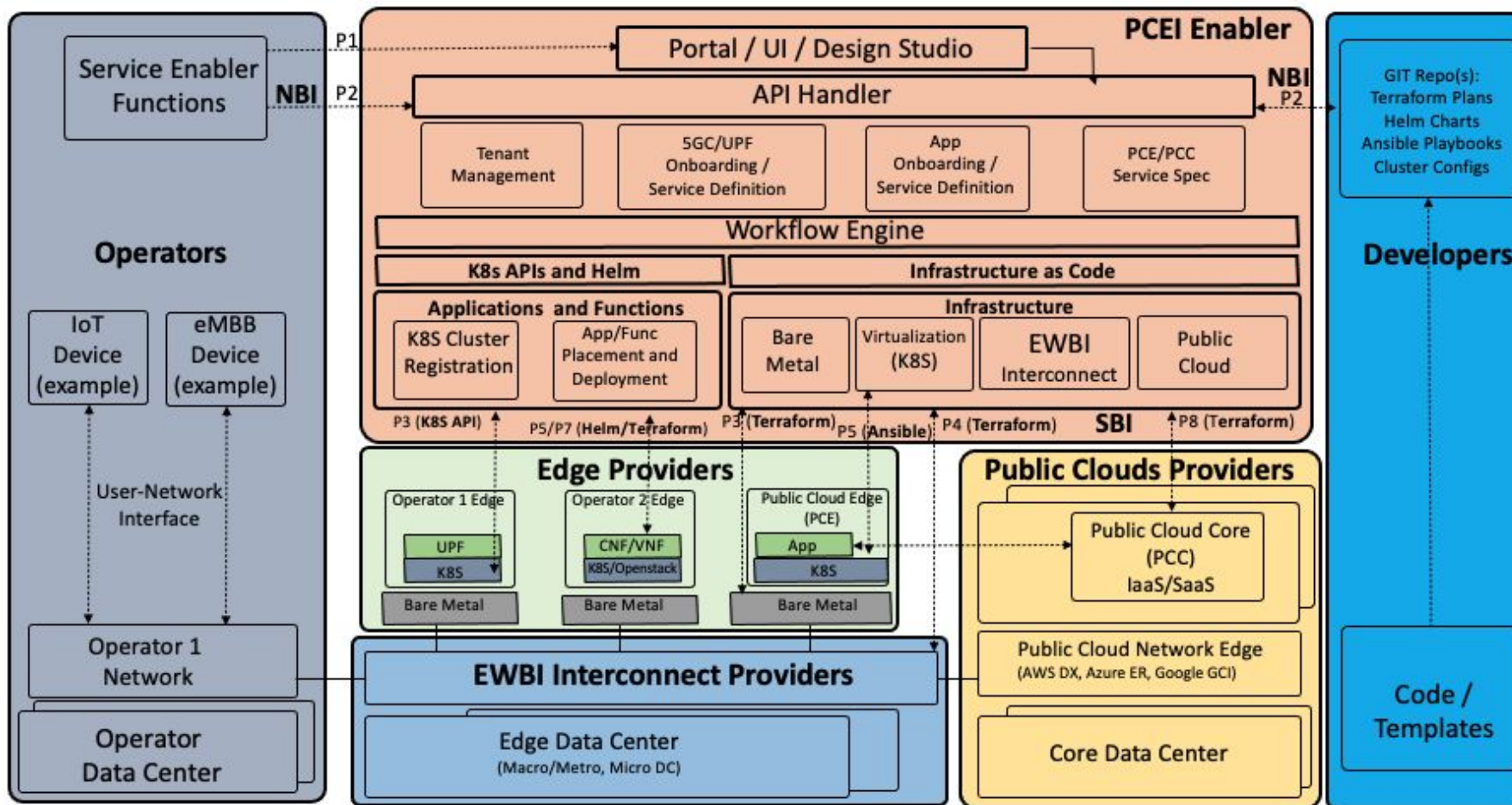
› Purpose

- › The purpose of Public Cloud Edge Interface (PCEI) Blueprint is to develop a **set of open APIs, orchestration functionalities and edge capabilities** for enabling Multi-Domain Interworking across the Operator Network Edge, the Public Cloud Core and Edge, the 3rd-Party Edge as well as the underlying infrastructure such as Data Centers, Compute Hardware and Networks.

› Terminology

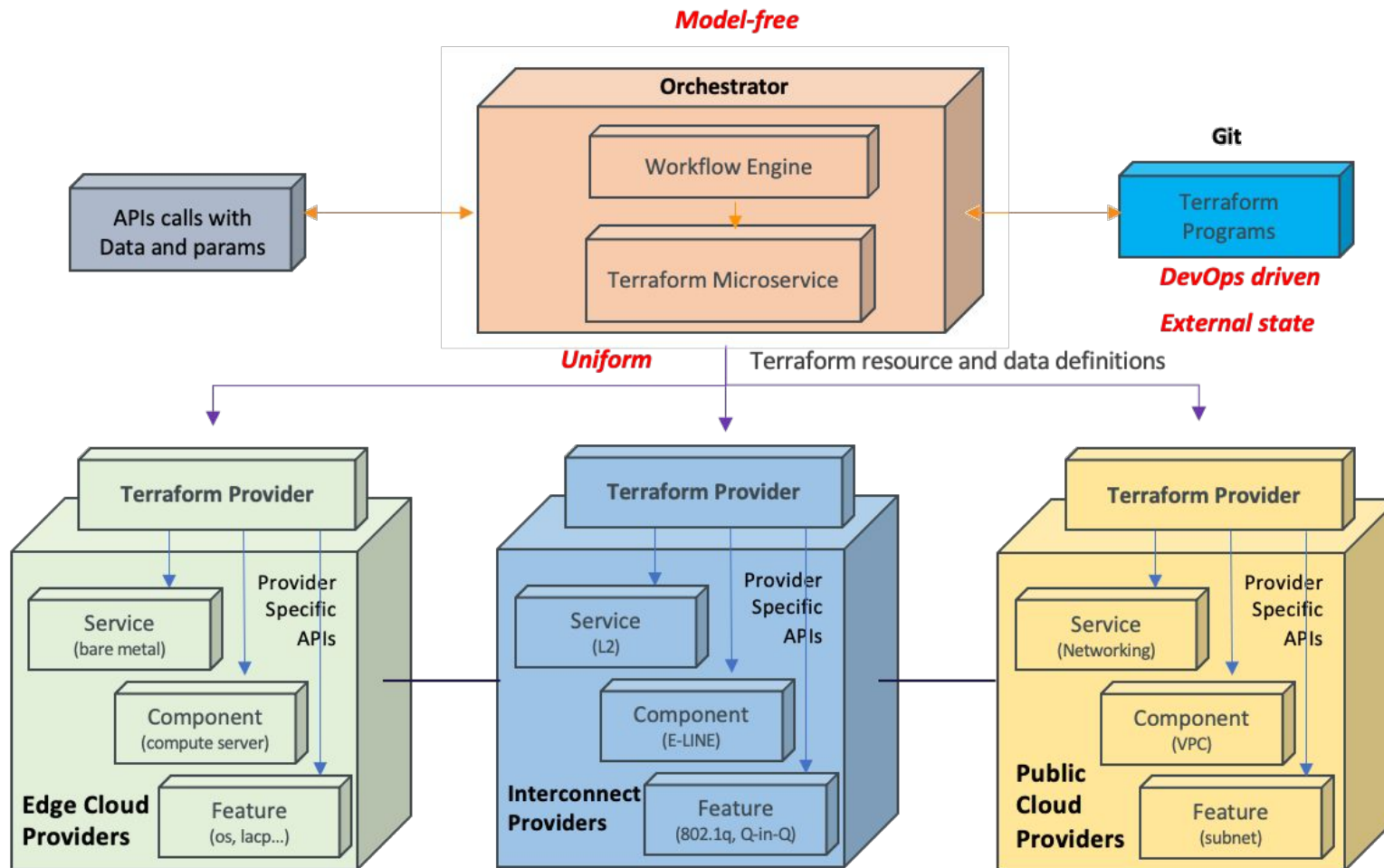
- › **Public Cloud Core (PCC)** – SaaS/IaaS resources, functions and services provided by Public Clouds.
- › **Public Cloud Edge (PCE)** - A set of Public Cloud resources, functions and applications running at the edge and coupled to PCC.
- › **Third-Party Edge (3PE)** – A set of resources, functions and applications running at the edge independently (de-coupled) from PCC.
- › **Operator Network Edge (ONE)** – A set of resources and functions provided by [Mobile] Network Operators and enabling user access to 3PE, PCE and PCC.

Akraino PCEI Release 6 Overview



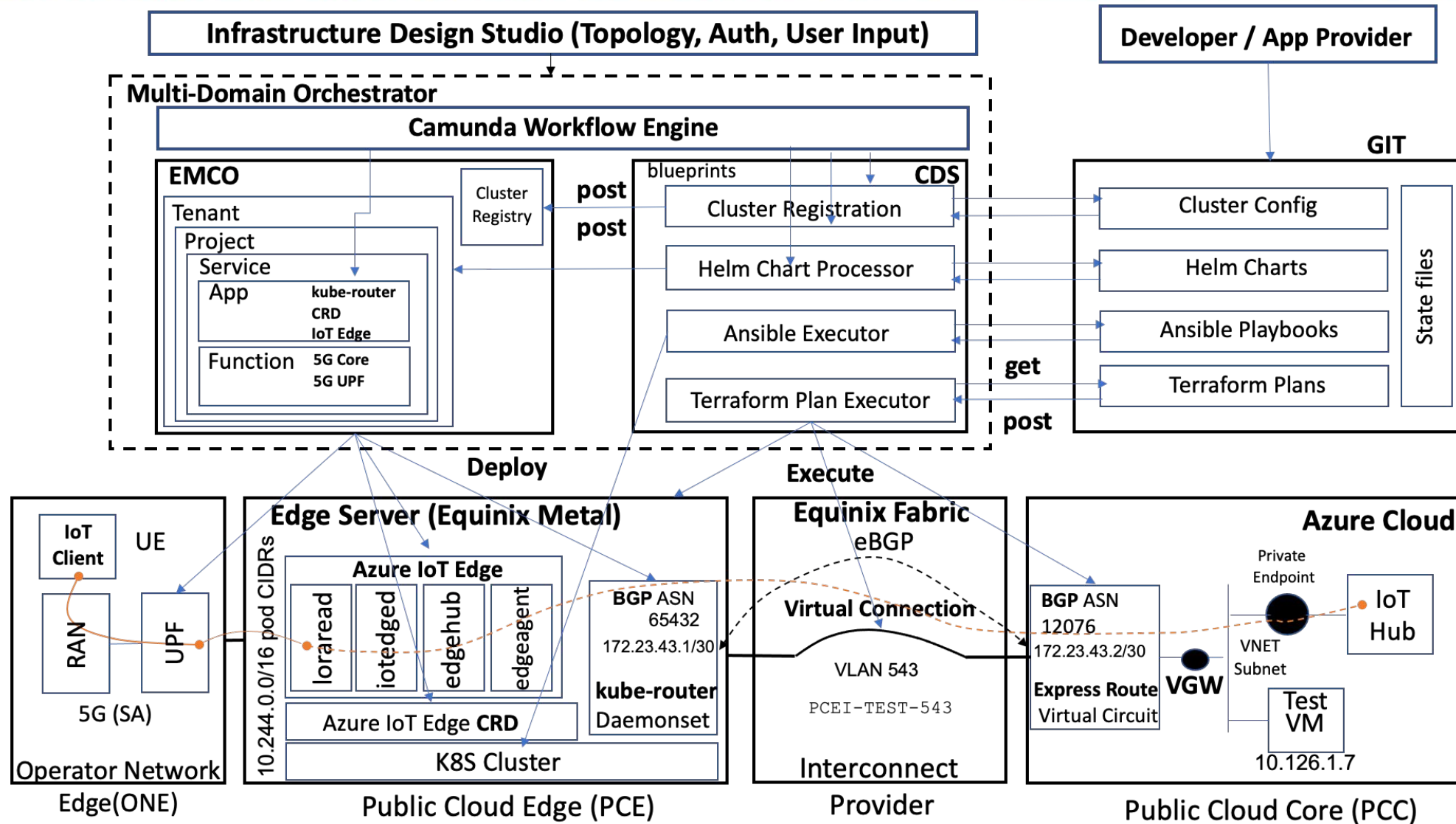
- ❑ NBI APIs
 - ❑ GIT Integration
 - ❑ Dynamic Edge Cluster Registration
 - ❑ Dynamic App Helm Chart Onboarding
 - ❑ Automatic creation of Service Instance in EMCO and deployment of Apps
 - ❑ Automatic Terraform Plan Execution
- ❑ Workflow Engine
 - ❑ Camunda
- ❑ Integrated Terraform Plan Executor
 - ❑ Azure (PCC)
 - ❑ AWS (PCC)
 - ❑ Equinix Fabric (Interconnect)
 - ❑ Equinix Metal (Bare Metal Cloud)
 - ❑ Openstack (3PE)
- ❑ Equinix Fabric Interconnect
- ❑ Multi-Public Cloud Core (PCC) Orchestration
- ❑ Kubernetes Edge
- ❑ Openstack Edge
- ❑ Cloud Native 5G UPF Deployment

Leveraging Terraform for orchestrating MEC infrastructure

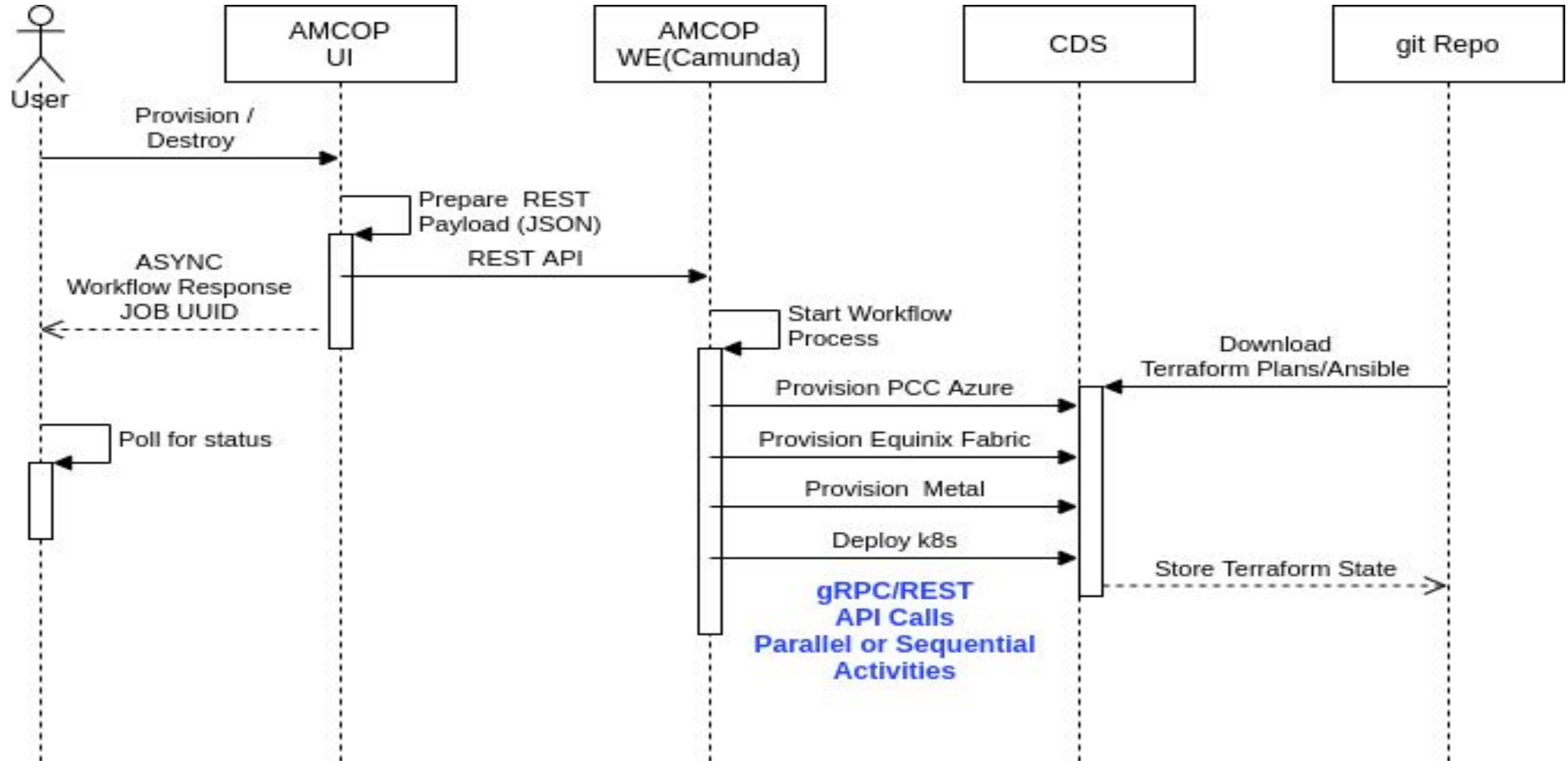


- **Uniform** - use of the same infrastructure orchestration methods across public clouds, edge clouds and interconnection domains.
- **Model-free** – the orchestrator does not need to understand the details of the individual infrastructure domains (i.e., implement their models). It only needs to know where to retrieve the Terraform plans for the domain in question and execute the plans using the specified provider.
- **DevOps driven** – the Terraform plans can be developed and evolved using DvOps tools and processes.
- **External state** – the state of infrastructure resources created by the orchestrator is stored outside of the orchestrator itself, making it stateless with respect to the infrastructure

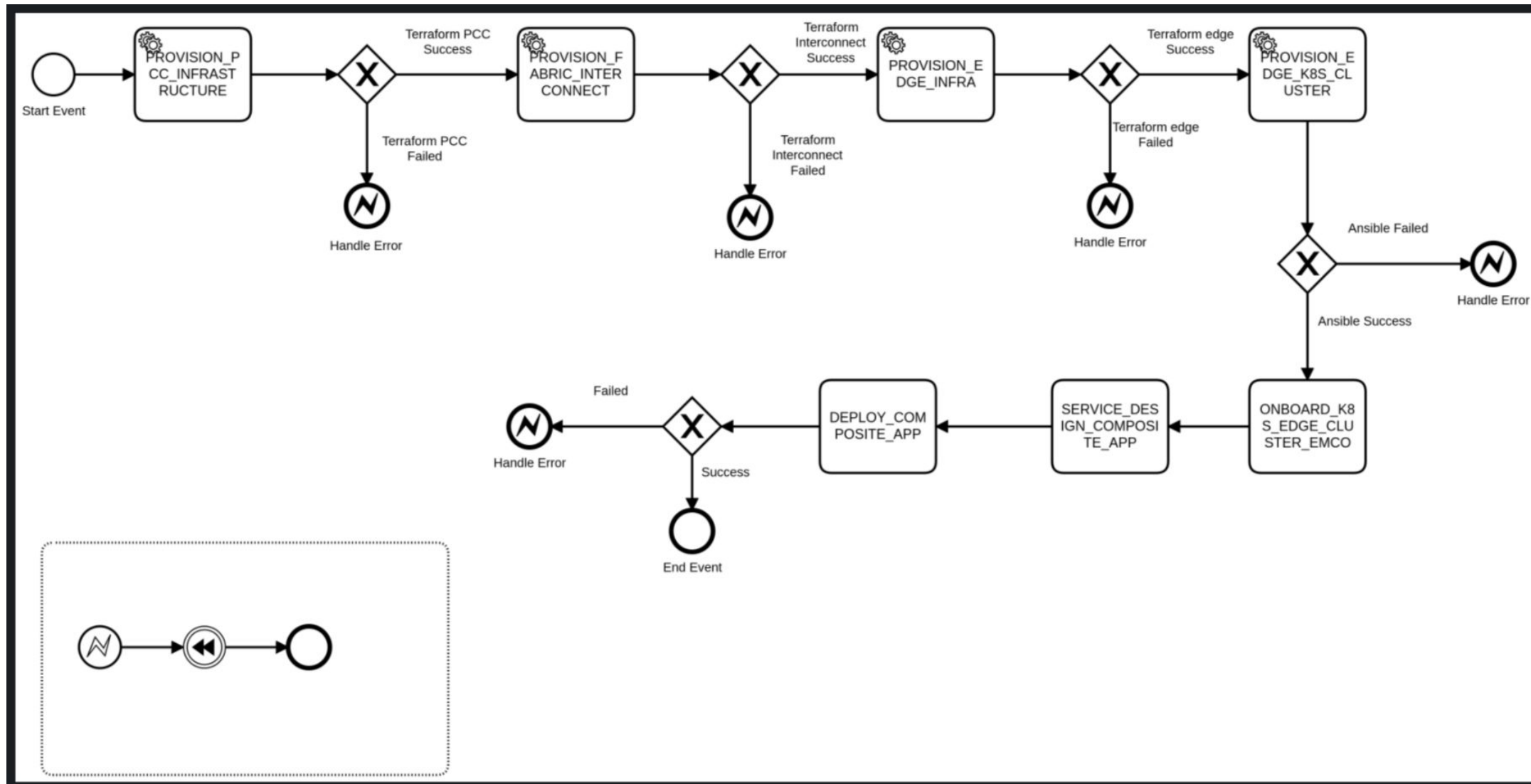
EMCO, ONAP CDS, Camunda & Terraform Integration Architecture



PCEI - Orchestration Workflow Sequence Diagram



Camunda workflow state machine



Demo : Edge to Cloud connectivity and Application Deployment

1. Deploy EMCO 2.0 (a new LFN project), CDS and CBAs
2. Developer Publish templates to GIT
 - a. Application Helm Charts (Azure IoT Edge, kube-router)
 - b. Terraform Plans
 - i. Azure Cloud
 - ii. Equinix Interconnect
 - iii. Equinix Metal
 - c. Ansible K8S (KUD) playbooks
3. Design Infrastructure
 1. Edge Cloud, Public Cloud, Interconnect
5. Provision Infrastructure (CDS/Terraform)
 1. Bare Metal (Equinix Metal Cloud)
 2. K8S on Bare Metal
 3. Azure Cloud (Express Route, Peering, VNET, VM, IoT Hub)
 4. Interconnect Edge Cloud with Public Cloud (Equinix Fabric)
6. Deploy Edge Application (PCE)
 1. Dynamic K8S Cluster Registration to EMCO
 2. Dynamic onboarding of App Helm Charts to EMCO
 3. Composite cloud native app (Azure IoT Edge, CRD, kube-router) deployment and end-to-end operation

Reference links

[PCEI R6 API Document](#)

[PCEI R6 Architecture Document](#)

[PCEI R5 Datasheet](#)

[PCEI R6 End-to-end Validation Guide](#)

[PCEI R6 Installation Guide](#)

[PCEI R6 Release Notes](#)

[PCEI R6 Test Document](#)