# **PCEI Use Case Development**

- User Plane Function Distribution and Local Break-Out
  - UPF Interconnection Scenario
  - UPF Placement Scenario
  - UPF Placement Options
    - API Structure
  - OPF Interconnect Options
    OPF Hardware Selection / Orchestration
  - UPF Configuration Orchestration Local Break-Out Options
  - 3PE/PCE Workload Instance Distribution
- Location Services
- Azure IoT Edge
  - PCEI Enabler IoT Edge Deployer Functions
  - PCEI Enabler IoT Edge Interconnect Functions
  - PCEI Enabler IoT Edge Software Distribution Functions
  - PCEI Enabler for Azure IoT Edge Implementation
    - Architecture and Interfaces
    - PCEI Enabler Structure
    - PCEI for Azure IoT Edge Call Flow (High-level)
    - Openstack HEAT Example
      - HEAT Template and Environment Files
      - Openstack HEAT Command

# User Plane Function Distribution and Local Break-Out

- UPF Distribution/Shunting capability -- distributing User Plane Functions in the appropriate Data Center Facilities with qualified compute hardware for routing the traffic to desired applications and network/processing functions/applications.
- Local Break-Out (LBO) Examples: video traffic offload, low latency services, roaming optimization.

The UPF Distribution use case distinguishes between two scenarios:

- UPF Interconnection. The UPF/SPGW-U is located in the MNO network and needs to be interconnected on the N6/SGi interface to 3PE and/or PCE/PCC.
- UPF Placement. The MNO wants to instantiate a UPF/SPGW-U in a location that is different from their network (e.g. Customer Premises, 3rd Party Data Center)

### **UPF** Interconnection Scenario

#### Assumptions:

- MNO hosts UPF/SPGW-U in their network.
- MNO provisions all UPF functions.
- MNO may request UPF Interconnection for Sgi/N6 interface in a required Metro to 3PE/PCE via APIs (on P2)
- Bare Metal Orchestration Provider and Interconnect Provider expose Data Center Location / Metro to PCEI Enabler via APIs (on P3 and P4)
- 3PE and PCE providers expose Data Center Location / Metro to PCEI Enabler via APIs (on P5 and P8/P7)
- PCEI Enabler may request Bare Metal Orchestration for Distributed UPF via APIs (on P3)
- PCEI Enabler may request Interconnect for MNO UPF (L2/L3) via APIs (on P4)
- PCEI Enabler may request PCC/PCE and/or 3PE Controller to instantiate workload instances for LBO processing via APIs (on P8/P5)



#### UPF Interconnection (N6/SGi) Call Flow Sketch

#### Notes:

- Does not include interconnect to 3PE/PCE/PCC to be added
- Does not include 3PE HW Orchestration to be added
- Does not include UPF LBO Orchestration to be added
- Does not include 3PE/PCE Workload Orchestration to be added

# PCEI Facilitated UPF Interconnection on N6/SGi Interface



### **UPF** Placement Scenario

Assumptions:

- MNO supports CUPS (5G NSA) and/or UPF (5G SA).
- MNO may request UPF Placement / Metro, and 3PE/PCE Access / Metro, via APIs (on P2)
- Bare Metal Orchestration Provider and Interconnect Provider expose Data Center Location / Metro to PCEI Enabler via APIs (on P3 and P4)
- 3PE and PCE providers expose Data Center Location / Metro to PCEI Enabler via APIs (on P5 and P8/P7)
- PCEI Enabler may request Bare Metal Orchestration for Distributed UPF via APIs (on P3)
- PCEI Enabler may request Bare Metal Orchestration for 3PE via APIs (on P3)
- PCEI Enabler may request Interconnect for Distributed UPF N3/S1 and N6/SGi traffic (L2/L3) via APIs (on P4)
- PCEI Enabler may expose UPF management access to MNO
- MNO may provision the Distributed UPF over management access
- PCEI Enabler may provision UPF connectivity and LBO configuration (based on the UPF provisioning model) over P2' using appropriate protocols
- PCEI Enabler may request PCC/PCE and/or 3PE Controller to instantiate workload instances for LBO processing via APIs (on P8/P5)



UPF Placement Call Flow Sketch

#### Notes:

- Does not include interconnect to 3PE/PCE/PCC to be added
- Does not include 3PE HW Orchestration to be added
- Does not include UPF LBO Orchestration to be added
- Does not include 3PE/PCE Workload Orchestration to be added

# PCEI Facilitated UPF Placement with N3(S1) Interconnect



### **UPF** Placement Options



## **API Structure**

To be added

## **UPF** Interconnect Options



UPF Hardware Selection / Orchestration



UPF Configuration Orchestration - Local Break-Out Options



## **3PE/PCE** Workload Instance Distribution

To be added

# **Location Services**

• Location Services -- location of a specific UE, or identification of UEs within a geographical area, facilitation of server-side application workload distribution based on UE and infrastructure resource location.

#### Assumptions:

- MNO provides a Location Service (LS) compliant with OMA Zonal Presence API (OMA-TS-REST\_NetAPI\_ZonalPresence)
- MNO may expose the Location Service via the Network Exposure Function (NEF)
- A Public Cloud Edge (PCE) instance is associated with a Zone (collection of Access Points such as small cells) provided by an MNO
- A 3rd-Party Edge (3PE) instance is associated with a Zone (collection of Access Points such as small cells) provided by an MNO
- An application/workload in the PCE requires Location Information (e.g. coordinates of the Access Point) for the UE/subscriber
- An application/workload in the 3PE requires Location Information (e.g. coordinates of the Access Point) for the UE/subscriber
- PCEI Enabler facilitates Zonal Presence API Request/Response routing between PCE and the MNO LS and between the 3PE and the MNO LS



# Azure IoT Edge

Assumptions:

- MNO hosts UPF/SPGW-U in their network.
- MNO provisions all UPF functions.
- MNO may request Deployment of IoT Edge on Bare Metal or NFVi via APIs (on P2)
- PCEI Enabler IoT Edge Deployer may request HW or Virtual resources (e.g. VM/Container) (on P3)
- PCEI Enabler IoT Edge Deployer may provision IoT Edge (install runtime, deploy standard modules, register with IoT Hub) via APIs/Deployer Code (on P8/P5)
- PCEI Enabler may request Interconnect for IoT Edge between MNO and Azure (L2/L3) via APIs (on P4)
- PCEI Enabler may request UPF LBO for MNO to direct customer traffic to IoT Edge (on P2/P2')
- PCEI Enabler may facilitate deployment of custom IoT Edge modules



## PCEI Enabler IoT Edge Deployer Functions

- Deploy a base Ubuntu VM
- Download and Install Microsoft GPG public key to apt config
- · Install moby engine and cli
- Install azure iot edge
- Modify iot edge config file using customer defined parameters (azure hub url, keys, scope id, etc)
- Deploy iot edge agent
- Deploy iot edge hub module
- Install certificates if edge is used as a gateway
- Restart iot edge

## PCEI Enabler IoT Edge Interconnect Functions

- Request virtual connectivity to MNO
- Request virtual connectivity to Azure (e.g. ExpressRoute)

# PCEI Enabler IoT Edge Software Distribution Functions

# PCEI Enabler for Azure IoT Edge Implementation

## Architecture and Interfaces



### **PCEI Enabler Structure**



### PCEI for Azure IoT Edge Call Flow (High-level)



### **Openstack HEAT Example**

#### **HEAT Template and Environment Files**

Template (pcei\_base.yaml)

#####

heat\_template\_version: 2013-05-23

description: Heat template that deploys PCEI IoT Edge VM in Openstack

## **# PARAMETERS #** ## parameters: pcei\_image\_name: type: string label: Image name or ID description: Image to be used for compute instance pcei\_flavor\_name: type: string label: Flavor description: Type of instance (flavor) to be used os\_private\_net\_id: type: string label: os management network name or ID description: Private network that connects os components and the VNF os\_private\_subnet\_id: type: string label: os management sub-network name or ID description: Private sub-network that connects os components and the VNF os\_private\_net\_cidr: type: string label: os private network CIDR description: The CIDR of the protected private network pcei\_private\_ip\_0: type: string label: VNF IP Address description: IP address that is assigned to the IoT Edge pcei\_name\_0: type: string label: VNF name description: Name of the vPacketGenerator key\_name: type: string label: Key pair name description: Public/Private key pair name pub\_key: type: string label: Public key description: Public key to be installed on the compute instance

# # # RESOURCES # # # ################

resources: random-str: type: OS::Heat::RandomString properties: length: 4

my\_keypair: type: OS::Nova::KeyPair properties: name: str\_replace: template: base\_rand params: base: { get\_param: key\_name } rand: { get\_resource: random-str } public\_key: { get\_param: pub\_key } save\_private\_key: false # Instance behing vRouter pcei\_private\_0\_port: type: OS::Neutron::Port properties: network: { get\_param: os\_private\_net\_id } fixed\_ips: [{"subnet": { get\_param: os\_private\_subnet\_id }, "ip\_address": { get\_param: pcei\_private\_ip\_0 }}] pcei\_0: type: OS::Nova::Server properties: image: { get\_param: pcei\_image\_name } flavor: { get\_param: pcei\_flavor\_name } name: { get\_param: pcei\_name\_0 } key\_name: { get\_resource: my\_keypair } networks: # - network: { get\_param: os\_private\_net\_id } - port: { get\_resource: pcei\_private\_0\_port } user\_data\_format: RAW user\_data: | #cloud-config password: pcei chpasswd: { expire: False } ssh\_pwauth: True runcmd: - [ sh, -xc, "sed -i 's,#UseDNS yes,UseDNS no,' /etc/ssh/sshd\_config" ] - systemctl restart sshd.service

Environment (pcei\_base.env)

#### ##########

parameters:

pcei\_image\_name: ubuntu1604

pcei\_flavor\_name: I3

os\_private\_net\_id: provider

os\_private\_subnet\_id: provider

os\_private\_net\_cidr: 10.121.11.0/24

pcei\_private\_ip\_0: 10.121.11.91

pcei\_name\_0: PCEI-IOT-EDGE

key\_name: maskey1

pub\_key: ssh-rsa AAAAB3NzaC1y

#### **Openstack HEAT Command**

openstack stack create -t pcei\_base.yaml -e pcei\_base.env