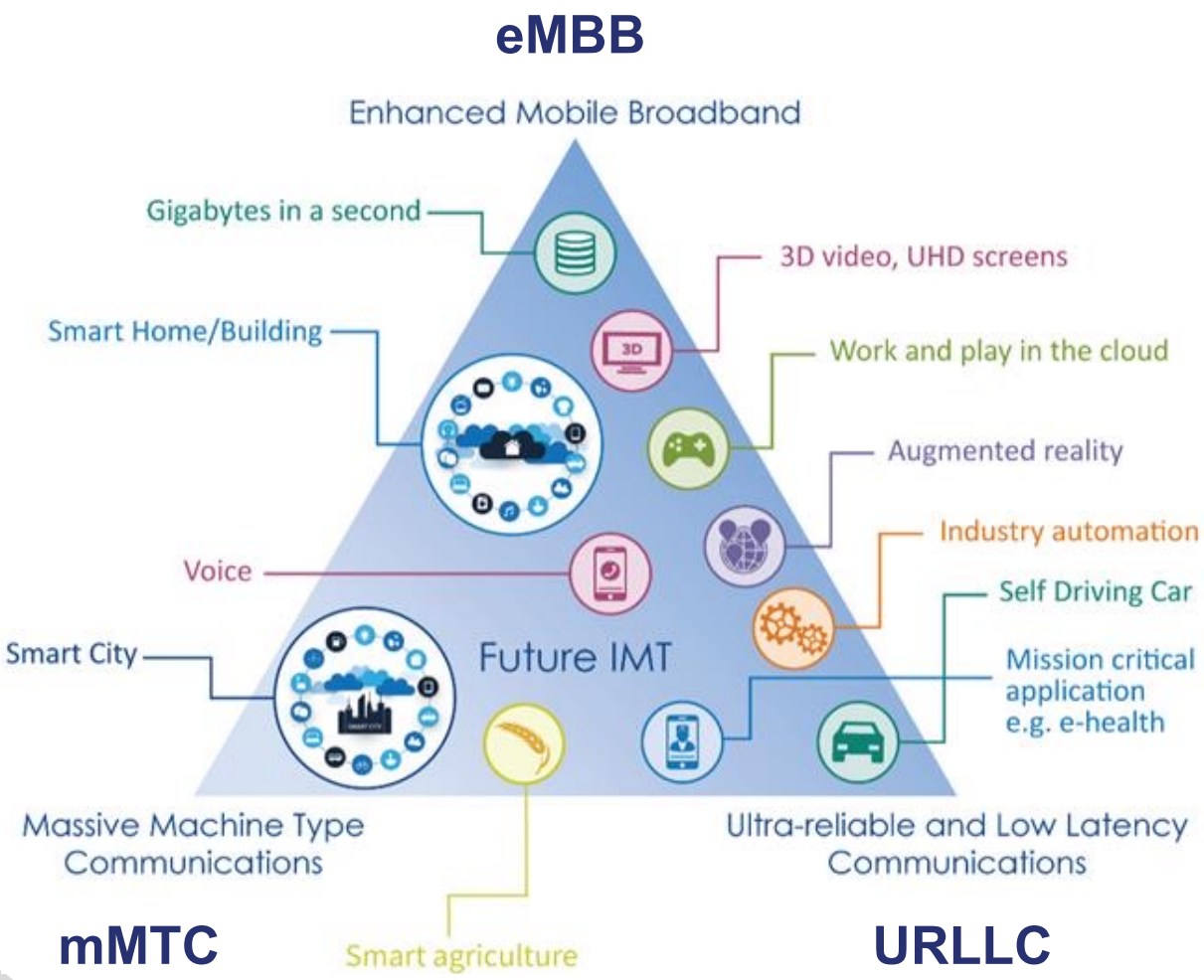
# IEEE P1935-based Smart Edge System

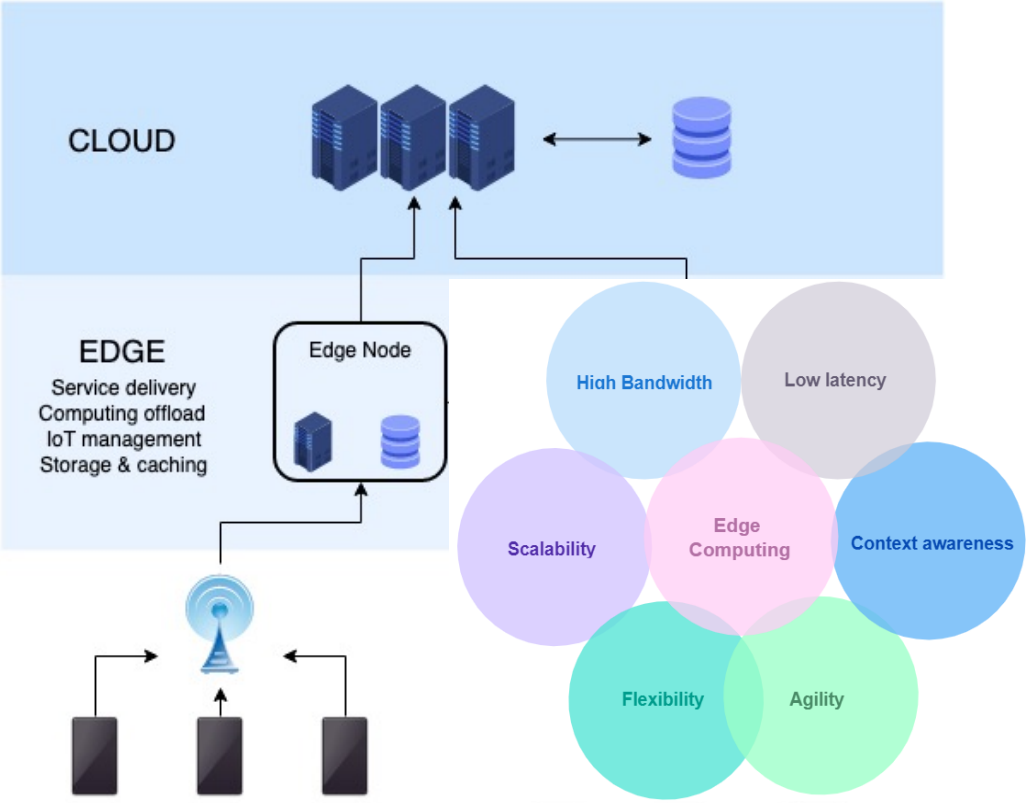
NTU Edge Team

**2022 Edge Computing Mini-workshop**

**2022.8.30**

# The Concepts and Advantages of Edge Computing



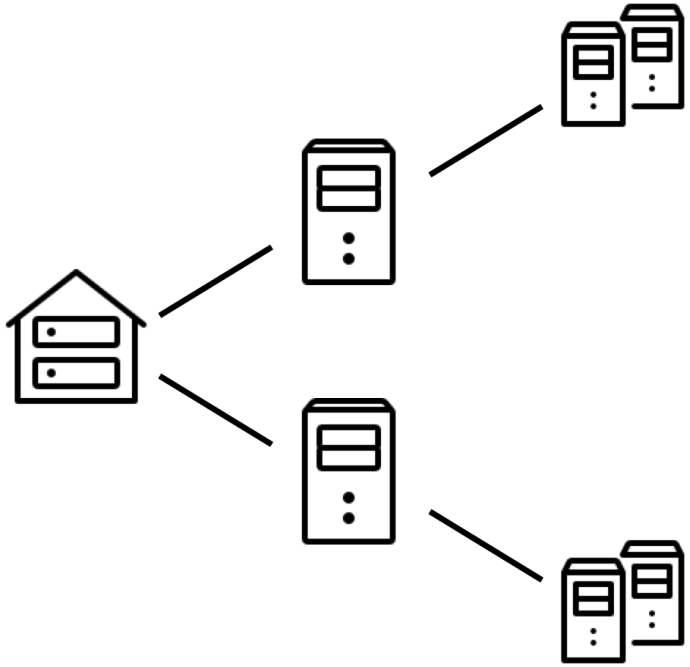
# Standard for Edge/Fog Management – P1935

- Introduced by the IEEE working group, P1935 define the
  - General architecture of the Edge/Fog system management and orchestration
  - Related service APIs involved in M&O of Edge/Fog platforms.
  - Lifecycle management and orchestration procedures for Edge/Fog applications

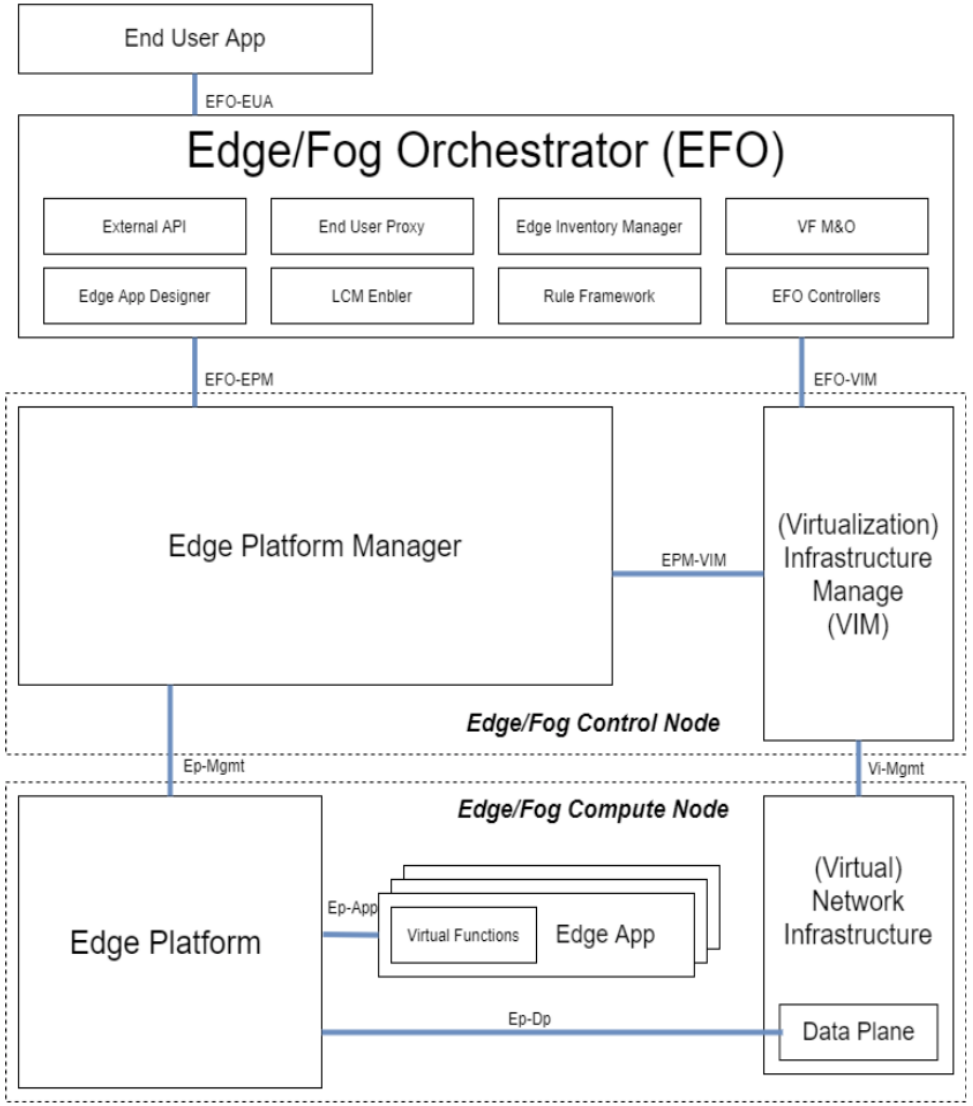
- Chair: Hung-Yu Wei

IEEE SA STANDARDS ASSOCIATION

# P1935 Standard Structure



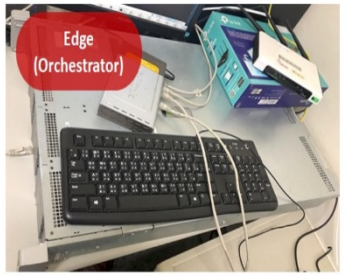
| Edge/Fog Orchestrator | Control Node | Compute Node |
|---|---|---|
| *An orchestrator of the whole Edge/Fog system* | *Manage the related compute nodes* | *An entity that is able to handle the computing tasks on its own* |



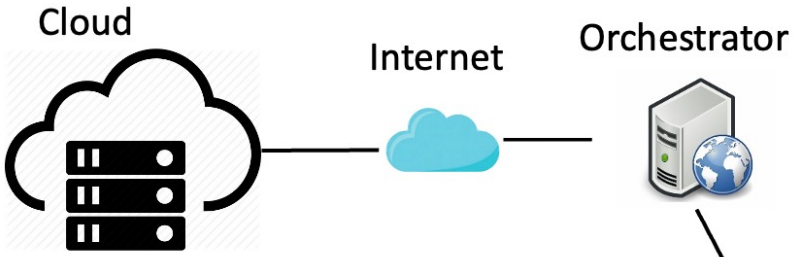
# SMART EDGE

## P1935 Smart Edge System

The system can reduce latency, improve service quality, and reduce traffic backhauled to the cloud. At the same time, through the designed algorithm to dynamically allocate computing resources, network resources, and storage resources, the performance of the applications running on it can be further improved.

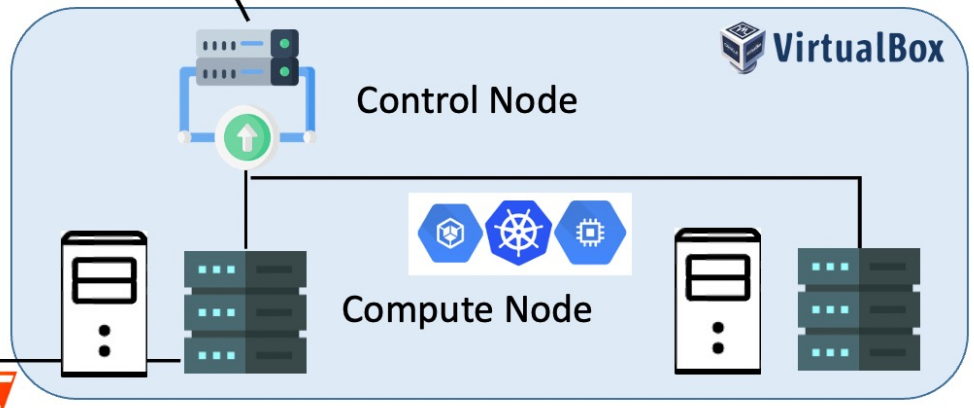



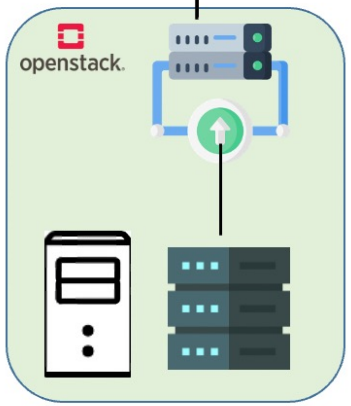


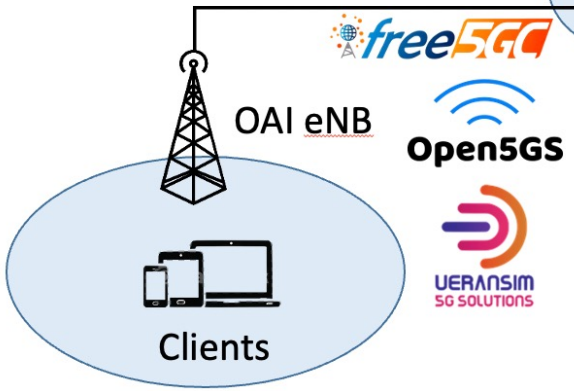



## Network Configuration

The smart edge system offers a variety of wired and wireless connection network setups, so it can be used in various scenarios.

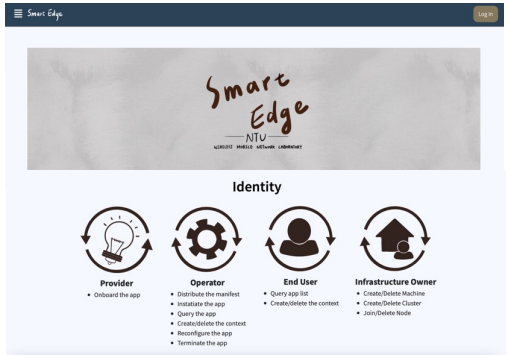


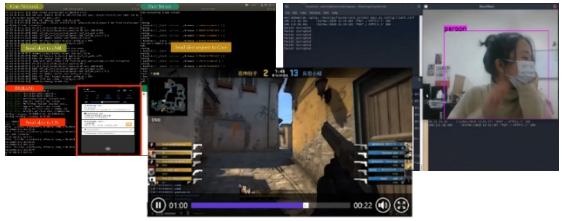




## UI / Web / Forum

We provide a web-based user interface. Users can operate the API of the P1935 and discuss various problems encountered on the edge platform.



AI Guard / Videos steaming / Facial Detection

## Edge-based Applications

Applications can enhance their performance through the integrated resource allocation provided by the system.

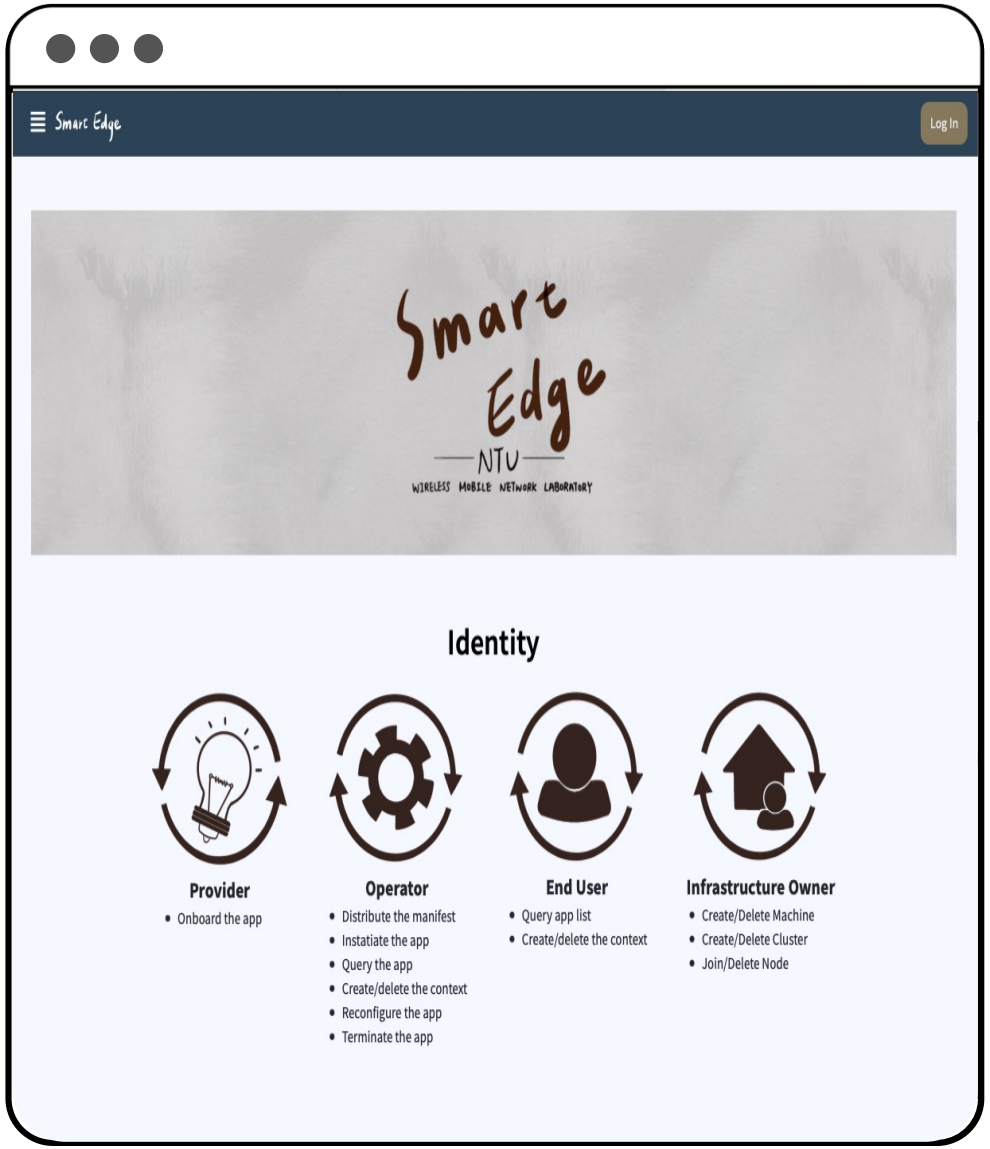
# Abstract

## P1935 Smart Edge system

- P1935 Standard
- Testbed Development
- P1935 Network Configuration and DDoS Detection
- UI & Websites
- Use Case Demo

## Testbed Researches

- Caching Related Work
- Create a over WAN cluster & Traffic Forwarding
- Scaling Problem on P1935 Smart Edge System



# P1935 Standard

Presenter: 陳則宇

# Table of Content of P1935 Standard

## Contents


1. Overview
    1.1 Scope
    1.2 Word usage

3. Architecture and Framework for Edge/Fog Manageability and Orchestration
    3.1 Edge/Fog Manageability and Orchestration Framework Overview
    3.2 Edge Orchestrator Level Entities
    3.3 Edge Controller Level Entities
    3.4 Edge Computer Level Entities
    3.5 External Entities
    3.6 Interfaces

4. Edge Platform Management and Orchestration
    4.1 Overview of Edge Platform Resource Management
    4.2 Resource Manageability Procedures

5. Edge Application Management and Orchestration
    5.1 Overview of Edge Application Management
    5.2 Manageability and Orchestration Procedures

6. Management Domains
    6.1 Domain Components
    6.2 Connectivity and Interoperability Domains (CID)
    6.3 Service and Application Domains (SAD)

# Clause 1: Overview

- This clause:
  - Introduces the background, development, and challenges of edge computing
  - Describes the purposes of this standard – management and orchestration of edge computing
  - Lists minimal requirements that an edge computing system shall meet

a) It shall be possible to deploy the Edge/Fog system in various positions in the wired, wireless, and mobile network, as well as the data center of the Edge Service Operators and Edge Service Providers.

b) It shall be possible to deploy the components, including the EFO, Edge/Fog control nodes, and Edge/Fog compute nodes, on the various hardware devices.

c) The Edge/Fog system shall fit in and communicate with the wireless communication network, dealing with the corresponding traffic routing to the correct components.

d) The Edge/Fog system shall be able to provide Edge services and Edge applications, as well as the corresponding environment for them.

e) The Edge/Fog system shall support the communications between the components and Edge services and Edge applications if authorized. This may include the different applications on different Edge/Fog compute nodes.

f) The Edge/Fog system shall support the full lifecycle and related operations of Edge applications, such as hosting, onboarding, instantiation, operating, and termination.

g) The Edge/Fog system shall support Edge Service Operators to manually and dynamically access the Edge services and Edge applications on their needs.

h) The Edge/Fog system shall be able to decide the service homing.

i) The Edge/Fog system shall support the compatibility with the authorized 3rd party entities, including processing the corresponding requests.

j) It shall be possible to deploy Edge applications on different Edge/Fog compute nodes without specific arrangement.

k) The Edge/Fog system shall be able to authorize and authenticate the user requests about the Edge applications.

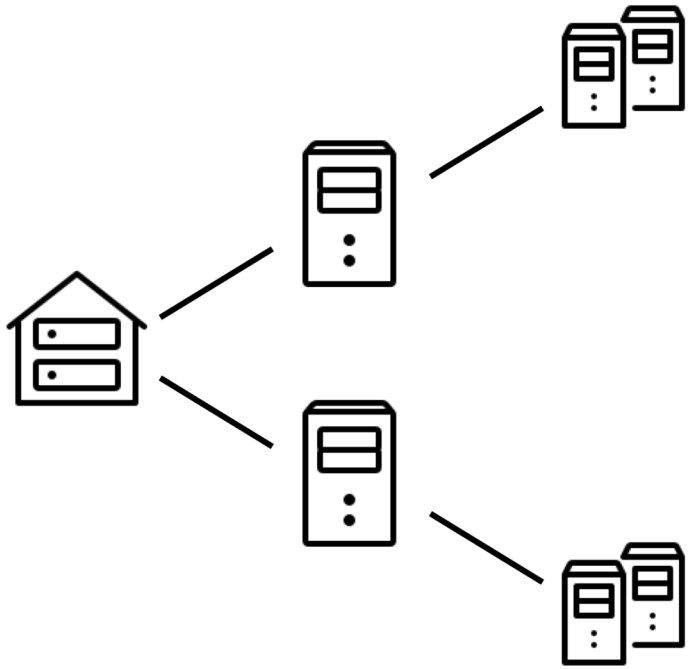
l) The Edge/Fog system shall support the service mobility, that is, keeping connectivity during the UE is moving from the coverage of an Edge/Fog control node to another.

m) The Edge/Fog system shall utilize the related network information to improve its own performance in various matrices.

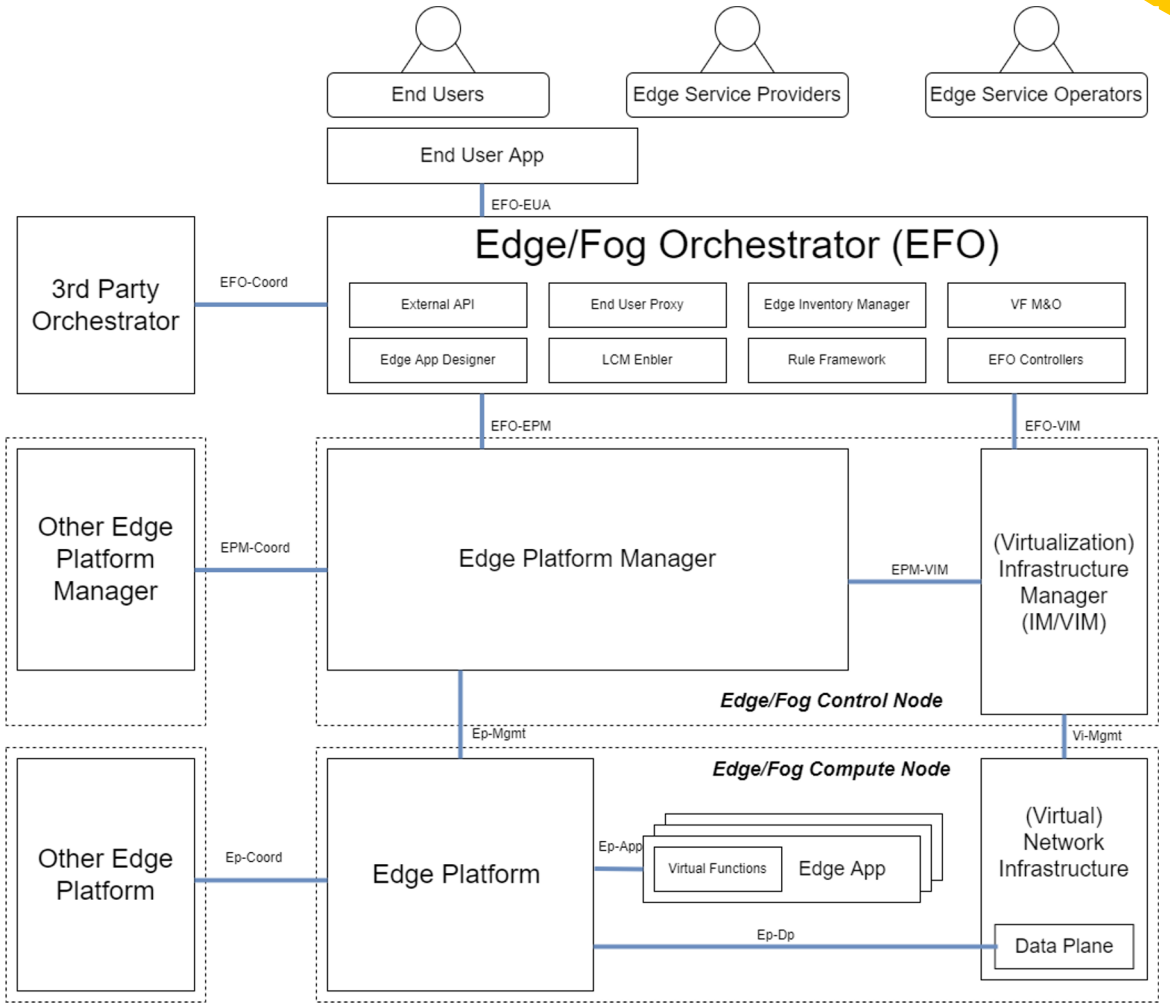
n) The Edge/Fog system shall support the Edge service to declare its own availability during service discovery.

o) The Edge/Fog compute nodes shall be able to provide computing, networking, and storage resources.

# Clause 3: P1935 Standard Structure

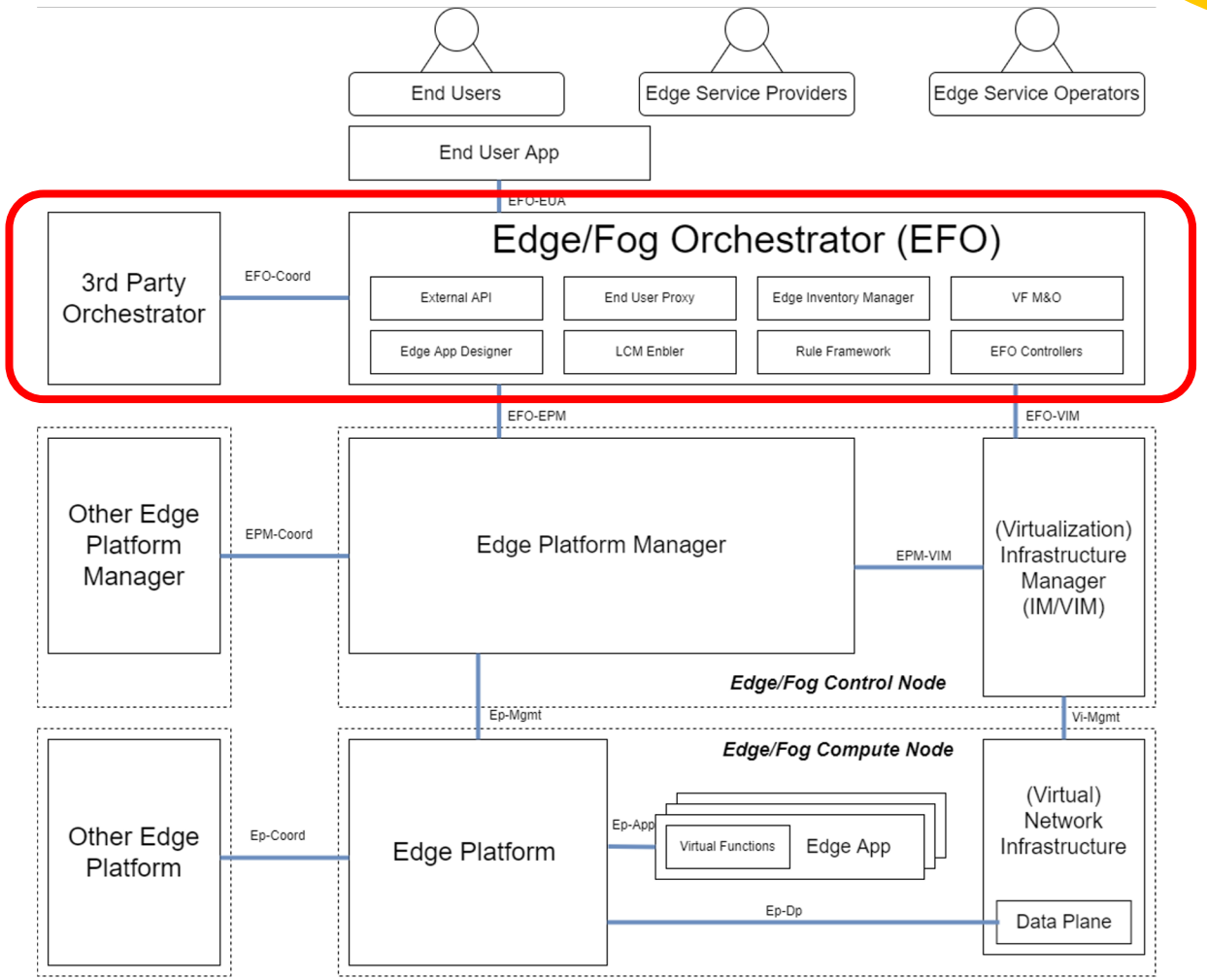


| Edge/Fog Orchestrator | Control Node | Compute Node |
|---|---|---|
| *An orchestrator of the whole Edge/Fog system* | *Manage the related compute nodes* | *An entity that is able to handle the computing tasks on its own* |



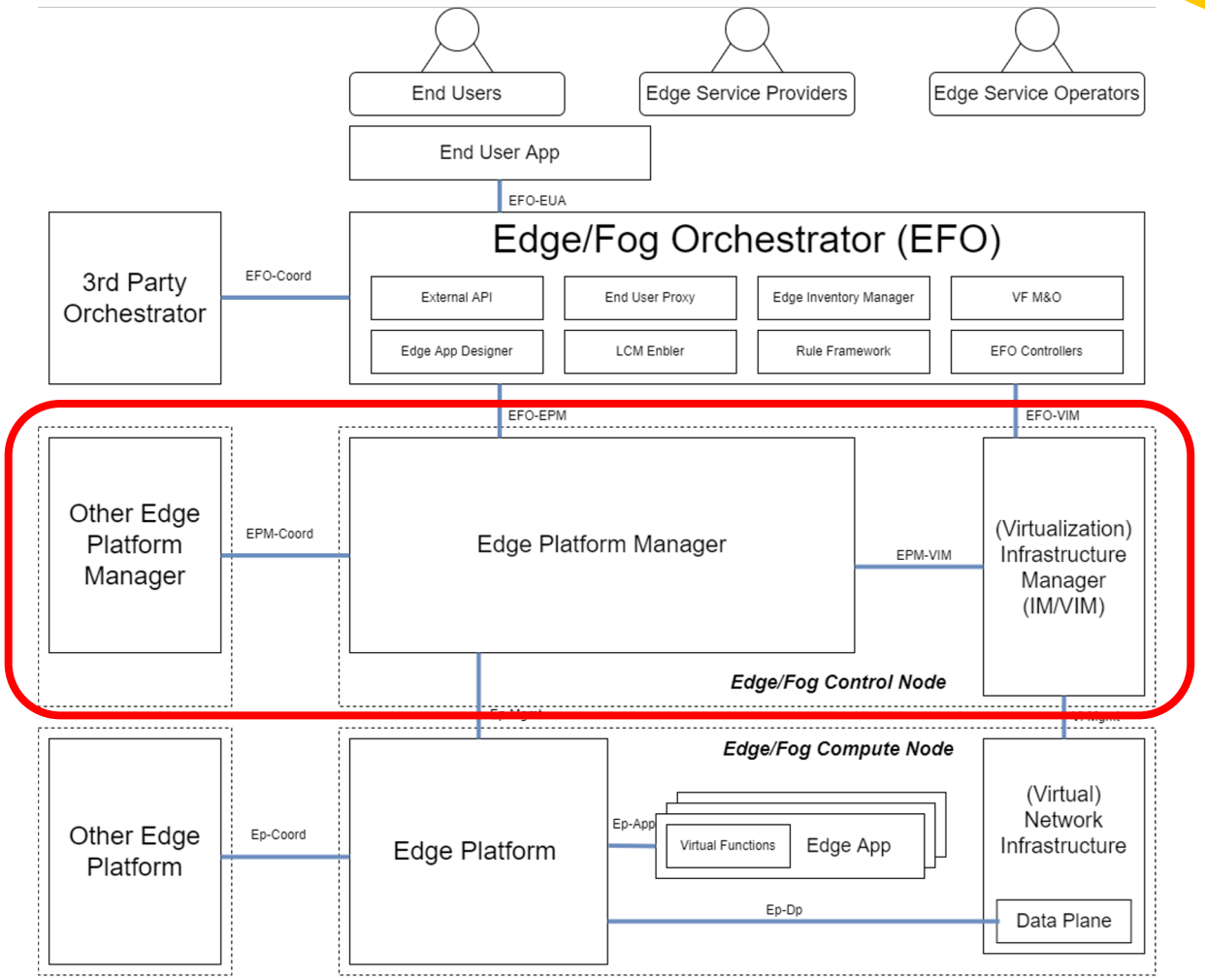
# Edge/Fog Orchestrator (EFO)

- Design time
  - Edge App Designer
  - Lifecycle Management Enabler
- Run time
  - Virtual Function Management and Orchestration
  - Rule Framework
  - Edge Inventory Manager
  - End User Proxy
  - EFO Controllers
- User Interface



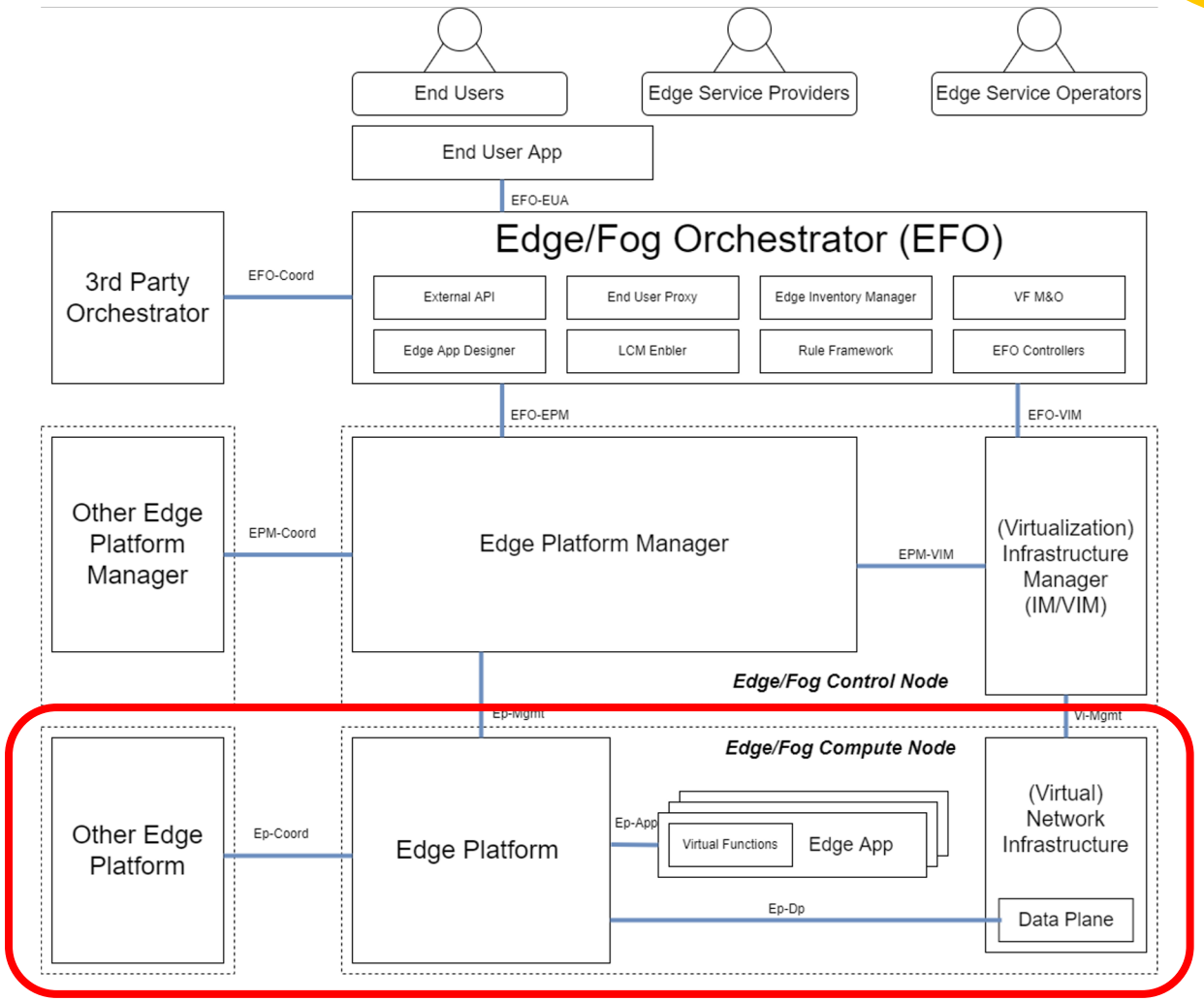
# Controller Level Entities

- Resource Management Elements
  - **Edge Platform Manager** (EPM)
- Application Management Elements
  - (Virtualization) **Infrastructure Manager** (IM/VIM)



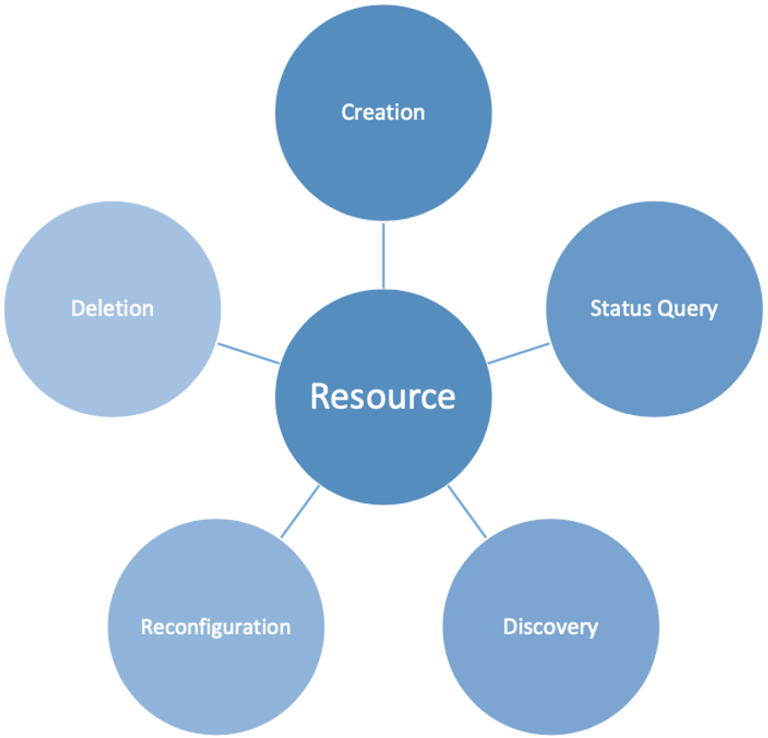
# Computer Level Entities

- Edge Platform
- Edge Application(s)
- Data Plane
- (Virtual) Network Infrastructure

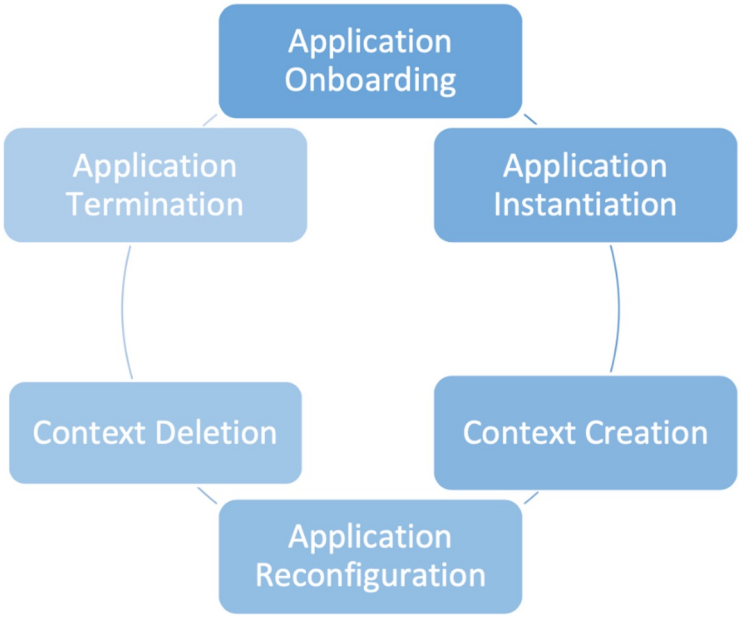


# IEEE P1935: Standard for Edge/Fog Manageability and Orchestration

Clause 4: Resource Management and Orchestration

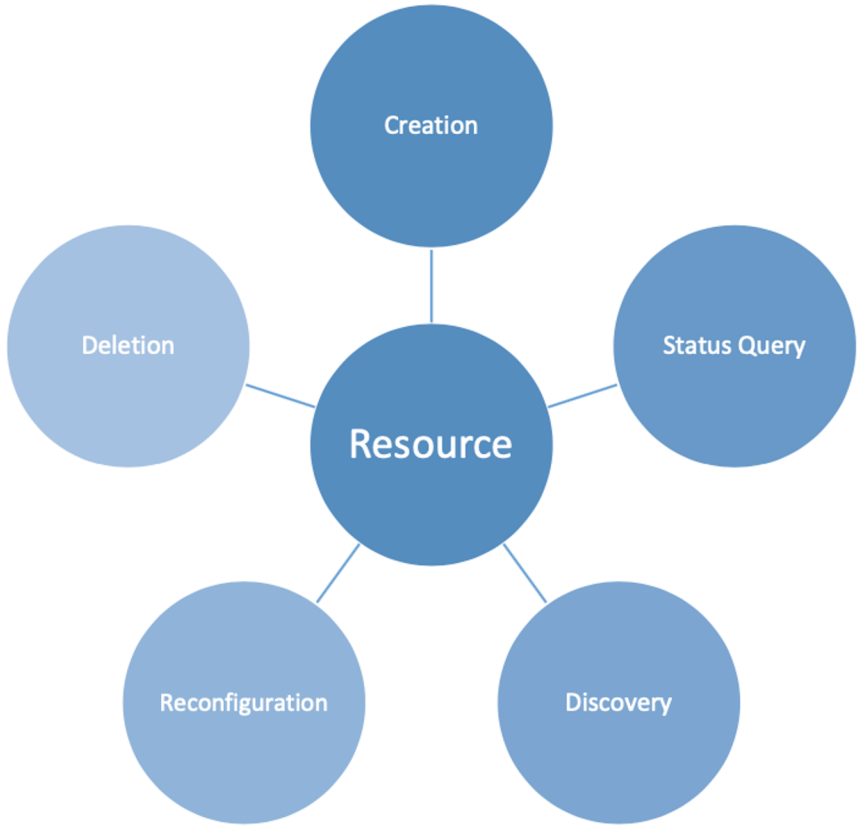



Clause 5: Application Management and Orchestration



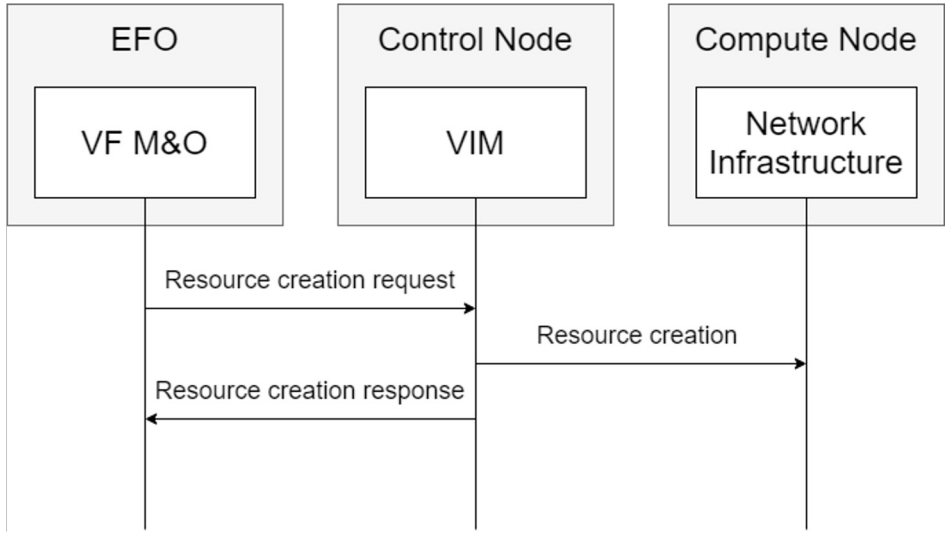
# Clause 4: Edge Platform Management and Orchestration

- Resource APIs
  - Resource Creation
  - Resource Status Query
  - Resource Discovery
  - Resource Reconfiguration
  - Resource Deletion
- Basic Specification of Resource APIs
  - The purpose of the API
  - Request format
  - Response format
  - HTTP methods supported
  - Representation supported



# Standard Content for Resource/App M&O

- Operation Description
- Conditions and Involved Information
- Operation Procedures (→)
- Requests and Responses (↓)




- **Table 7—Resource creation request**

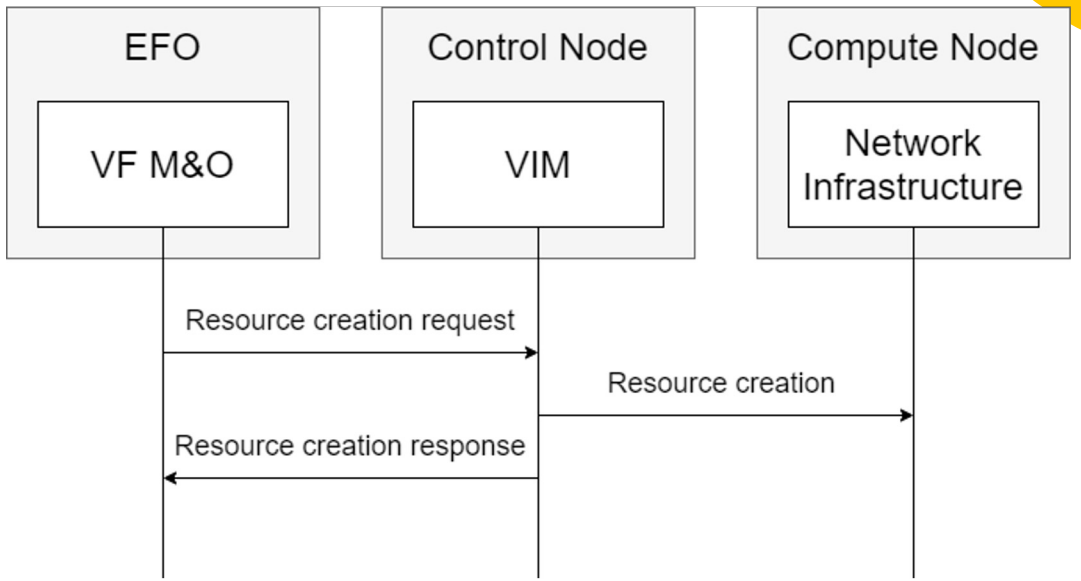
| Element | Description |
|---|---|
| Request Type | HTTP POST |
| Parent Resource ID | The identifier of the parent resource of the created resource. It shall be contained in the URL. |
| Request Payload | The payload contains the information and data of the created resource. |

- **Table 8—Resource creation successful response**

| Element | Description | |
|---|---|---|
| HTTP Status Code | 201 Created | |
| HTTP Header | URL | The URL of the created resource. |
| Response Payload | (Empty) | |

# Example 1: Resource Creation

- Operation Description
- Conditions and Involved Information
- Operation Procedures (→)
- Requests and Responses (↓)




- The pre-condition of the procedure:
  - The target resource is not yet created in the Edge system.
- The post-condition of the procedure:
  - The resource has been created and configured as network infrastructure in the Edge system.

- **Table 7—Resource creation request**

| Element | Description |
|---|---|
| Request Type | HTTP POST |
| Parent Resource ID | The identifier of the parent resource of the created resource. It shall be contained in the URL. |
| Request Payload | The payload contains the information and data of the created resource. |

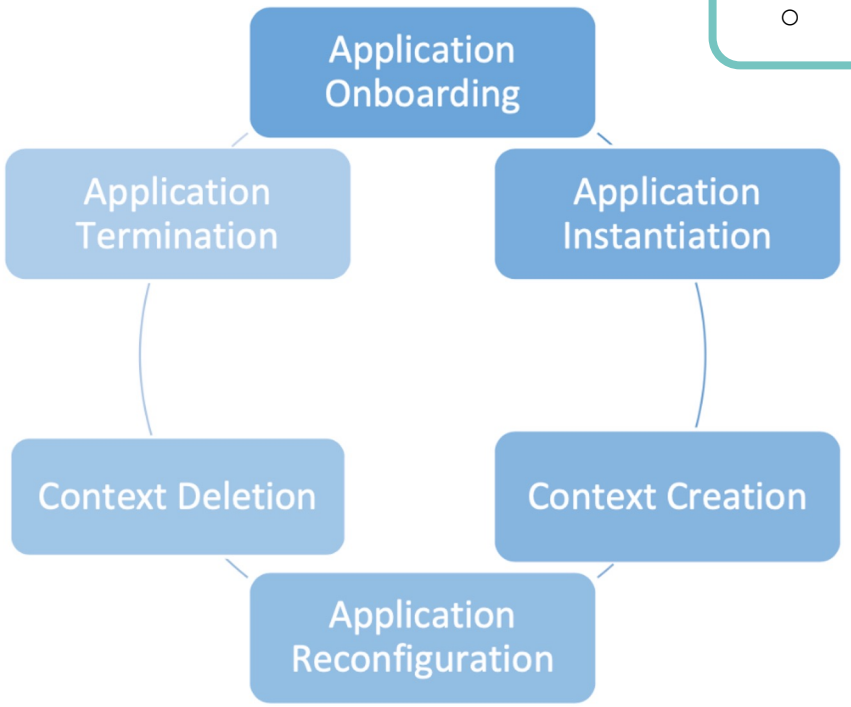
- **Table 8—Resource creation successful response**

| Element | Description | |
|---|---|---|
| HTTP Status Code | 201 Created | |
| HTTP Header | URL | The URL of the created resource. |
| Response Payload | (Empty) | |

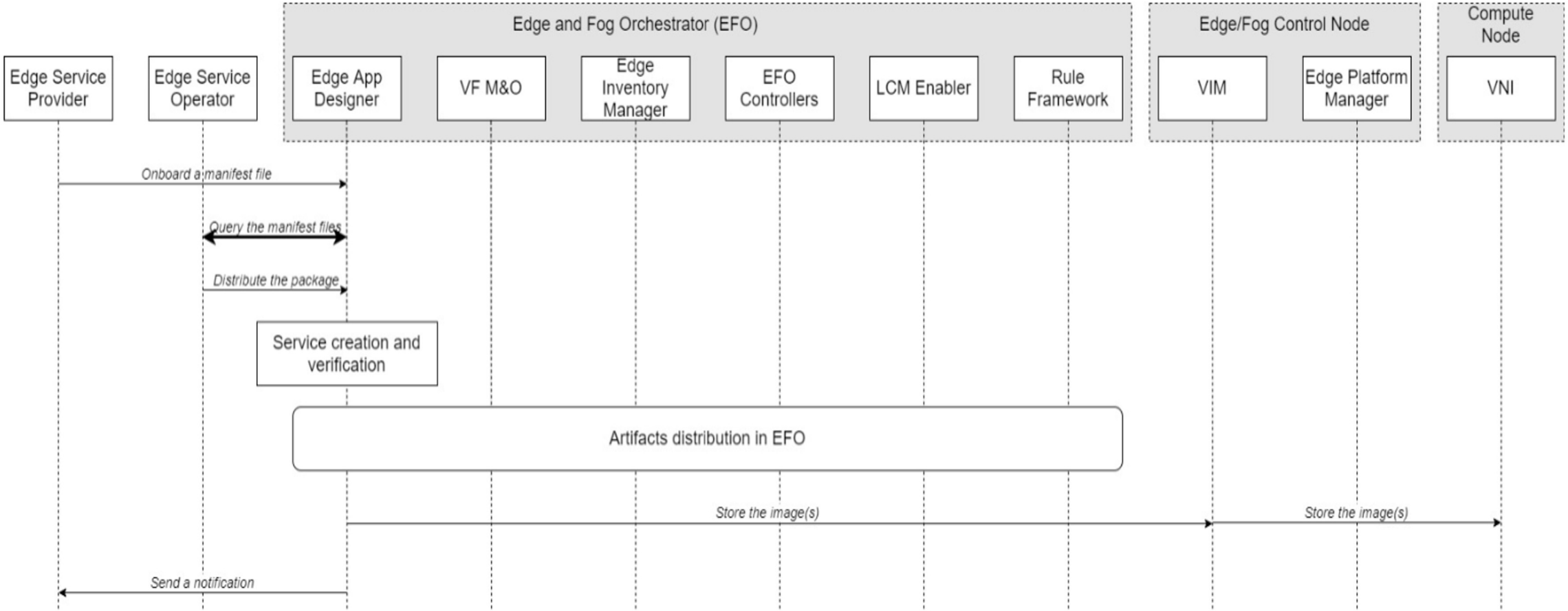
# Clause 5: Edge Application Management and Orchestration

- Lifecycle Management Procedures
  - App Onboarding
  - App Instantiation
  - Context Creation
  - App Reconfiguration
  - Context Deletion
  - App Termination

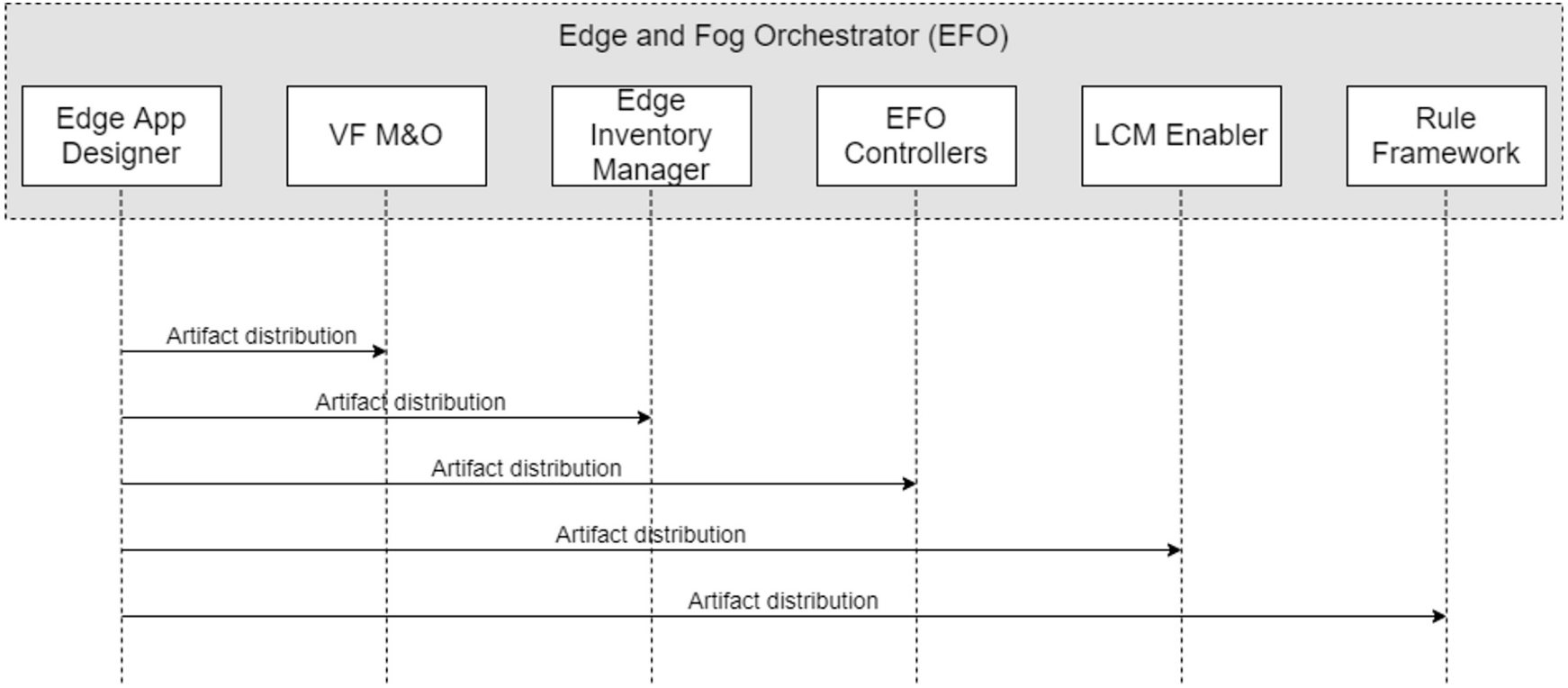
The files used in these procedures
- Application Context
- Application Package
- Artifact
- Blueprint File
- Inventory
- Manifest File
- VF Image



# Example 2: Application Onboarding

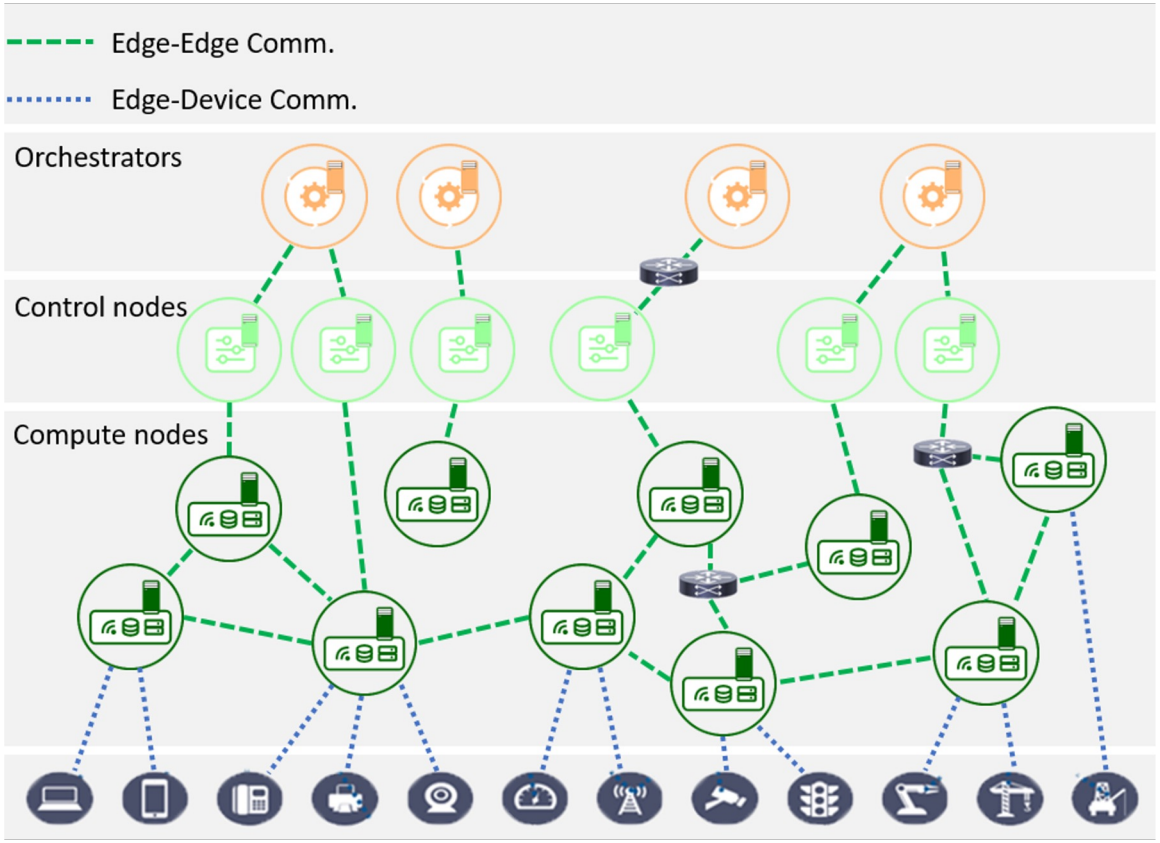


# Artifact Distribution in App Onboarding

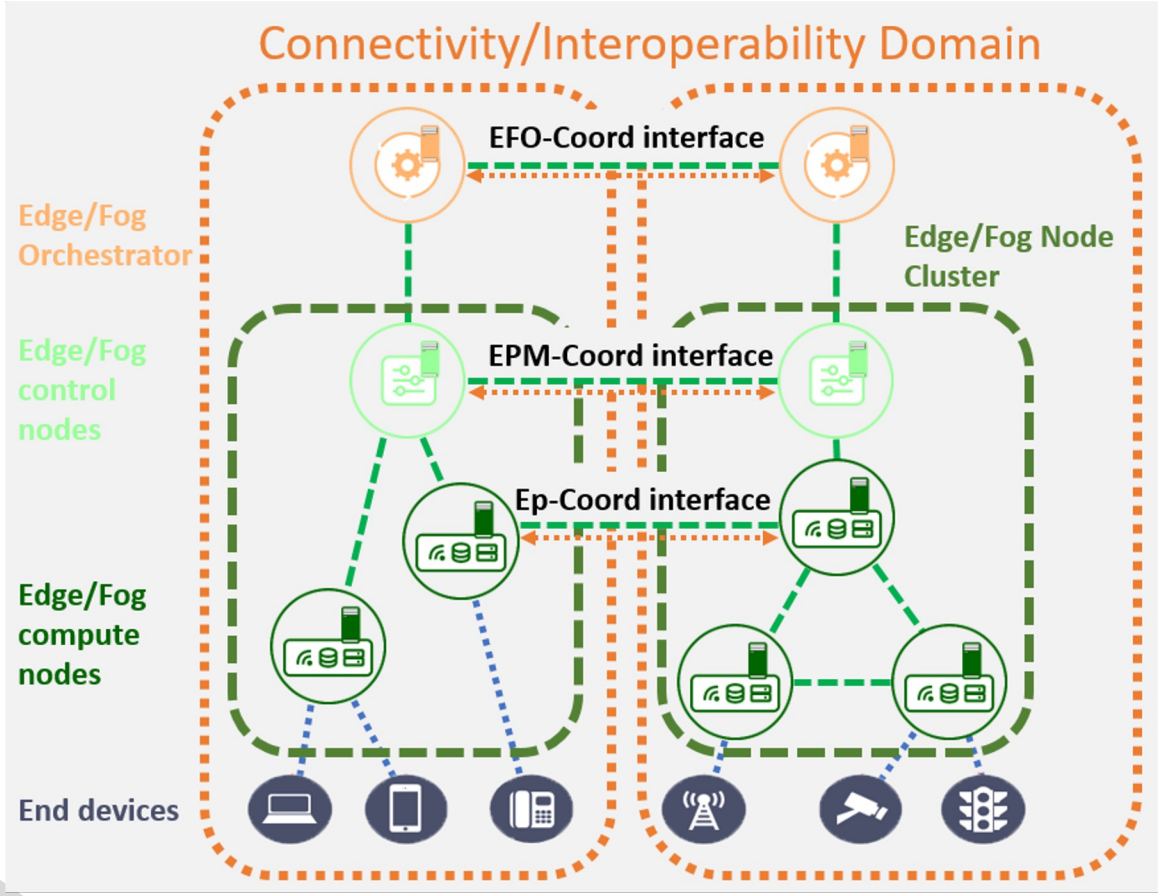


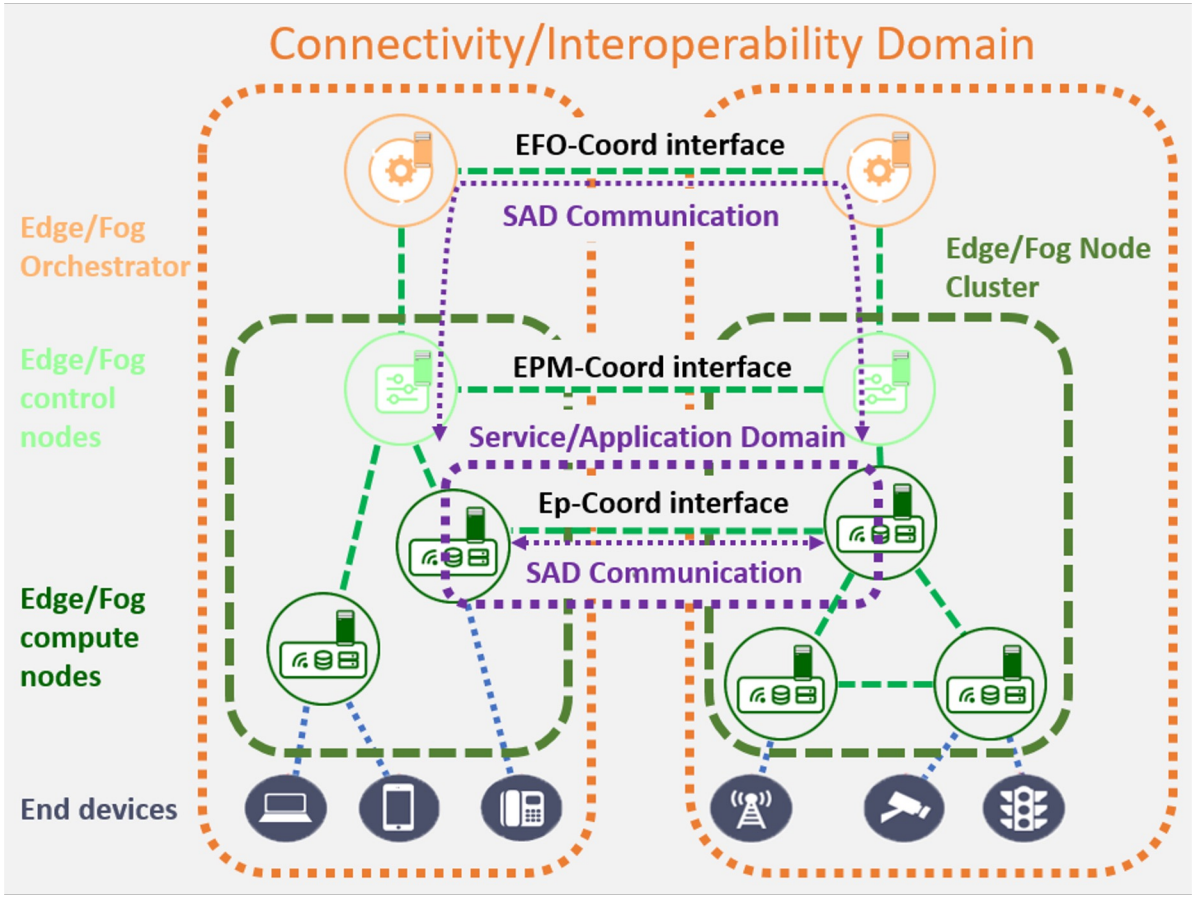
# Clause 6: Management Domains of P1935 Edge System

- This clause involves the relationship between multiple edge entities and even systems
  - Define the concept of
    - physical, logical, and virtual edge nodes
    - edge node clusters and border edge nodes
  - Introduce Connectivity and Interoperability Domains (CID) and Service and Application Domains (SAD)



# Connectivity/Interoperability Domain & Service/Application Domain



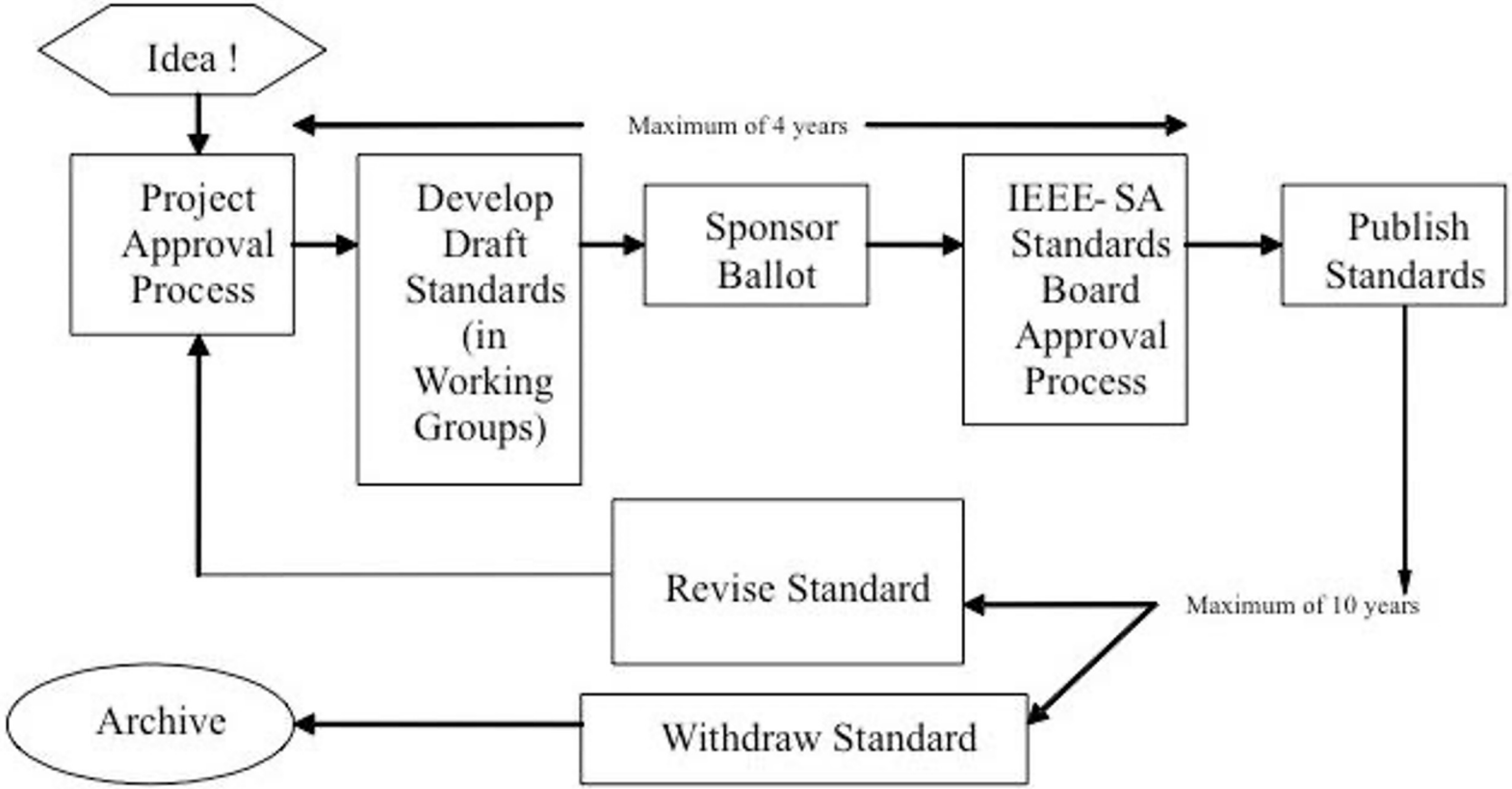




* Interconnect & Interoperate

# Conclusions

- We introduce the standard P1935, a standardized, orchestration-wise design to simplify the process and offer better system performance and user experience.
  - The system architecture is depicted in Clause 3
  - The resource management and application lifecycle management are described in Clause 4 and 5 respectively
  - The relationship between multiple edge systems is introduced in Clause 6

# Appendix A: IEEE Standard Ballot Process




Ref: https://standards.ieee.org/wp-content/uploads/import/visuals/infographics/stdsmade.jpg

# Appendix B: Requests and Responses

Table 21—Default format of a request/response

| Element | Description |
|---|---|
| Operation status | A 200 OK code shall be returned if successful.<br>A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed. |

5.2.1.3 Requests and Responses

The following tables describe the elements contained in the request.

Table 24—Application manifest file uploading request

| Element | Description |
|---|---|
| Service Provider ID | The Edge Service Provider identifier. |
| Manifest file | A manifest file that includes an application package and a blueprint file. It is the payload of the request. |

Table 25—Application manifest file uploading response

| Element | Description |
|---|---|
| Operation status | A 201 Created code shall be returned if successful.<br>A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed. |
| Manifest file ID | The generated identifier of the uploaded manifest file. This field will be blank if the upload fails. |

Table 26—Uploaded manifest files query request

| Element | Description |
|---|---|
| Operator ID | The Edge Service Operator identifier. The ID shall be authorized by the EFO. |
| Query filter | The filter conditions for the query. Operators can only query the manifest files that satisfy these conditions and limitations. |

Table 27—Uploaded manifest files query response

| Element | Description |
|---|---|
| Operation status | A 200 OK code shall be returned if successful.<br>A 400 Bad Request/401 Unauthorized/403 Forbidden code shall be returned if failed. |
| Query result | A list of the manifest files that satisfy the query conditions. |

Table 28—Artifact distribution request

| Element | Description |
|---|---|
| Operator ID | The Edge Service Operator identifier. The ID shall be authorized by the EFO. |
| Manifest file ID | The identifier of the manifest file which the Edge Service Operator supposes to distribute. |

Table 29—Distribution success notification

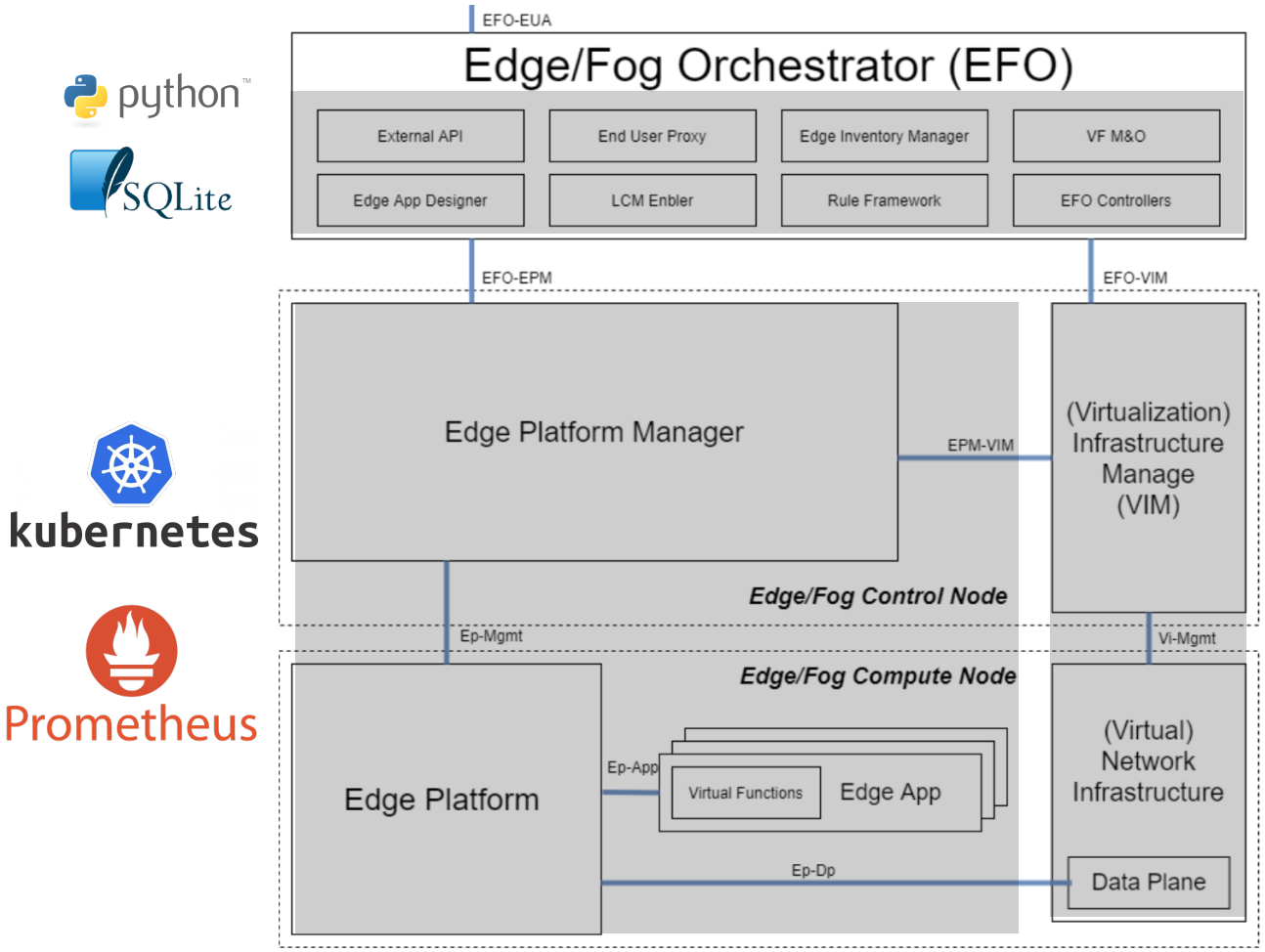
| Element | Description |
|---|---|
| Operator ID | The Edge Service Operator identifier. |
| Application ID | The packages are distributed successfully, and the application is given an identifier for the Edge Service Operators to instantiate it in the next subclause. |

# Testbed Development

Presenter: 陳奐廷

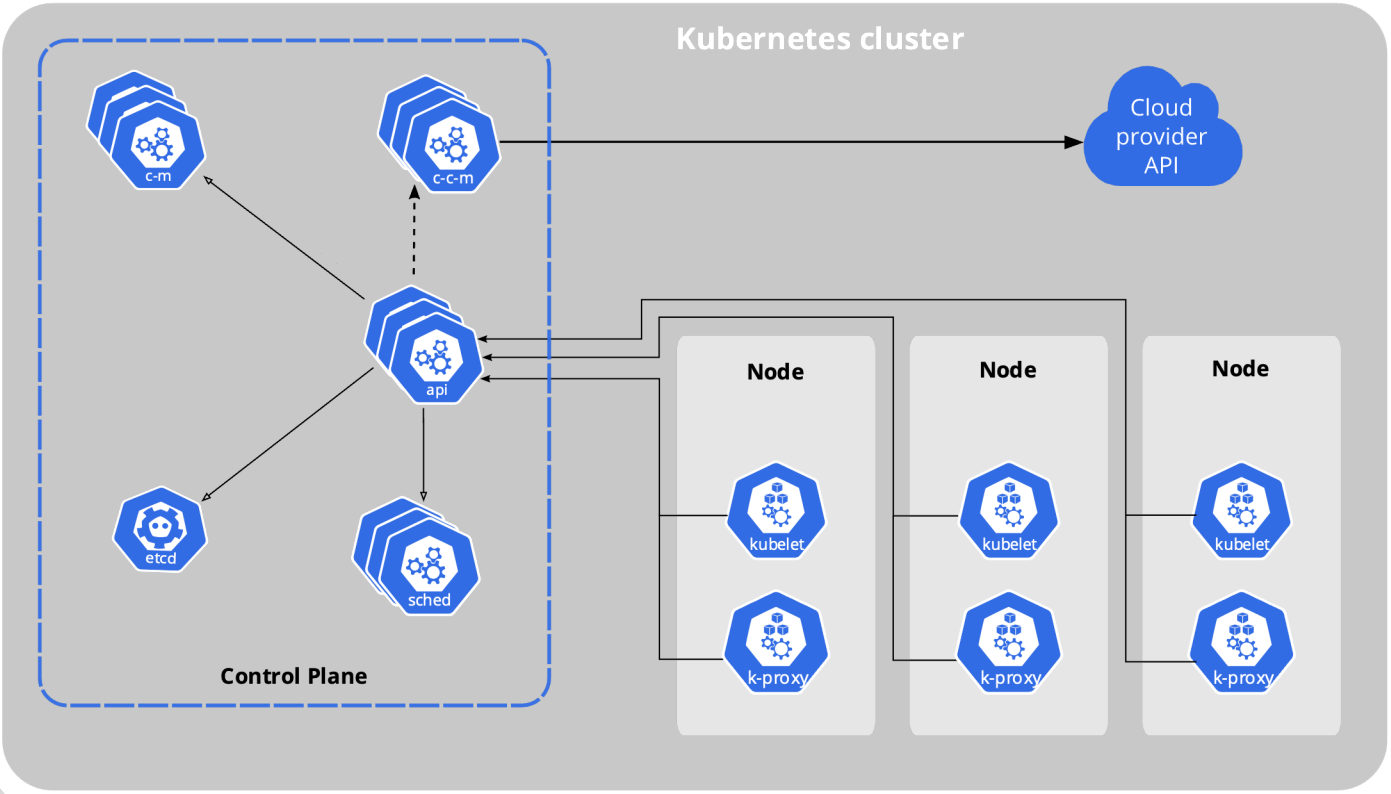
# Platform Development

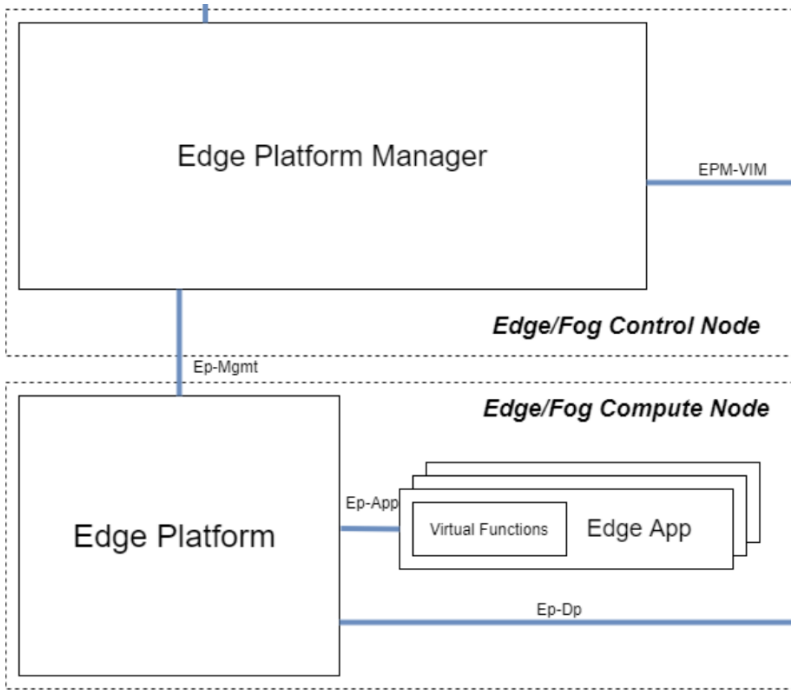
- Python
  - Expose APIs to user (Flask)
  - Manage control/compute node
  - Database (SQLite) to record user/node/application info
- Kubernetes
  - Server/App management
  - Scalability, Self-healing, …
- Prometheus
  - Monitor the machine status
- VirtualBox
  - Virtual machine management



# Tools Introduction - Kubernetes

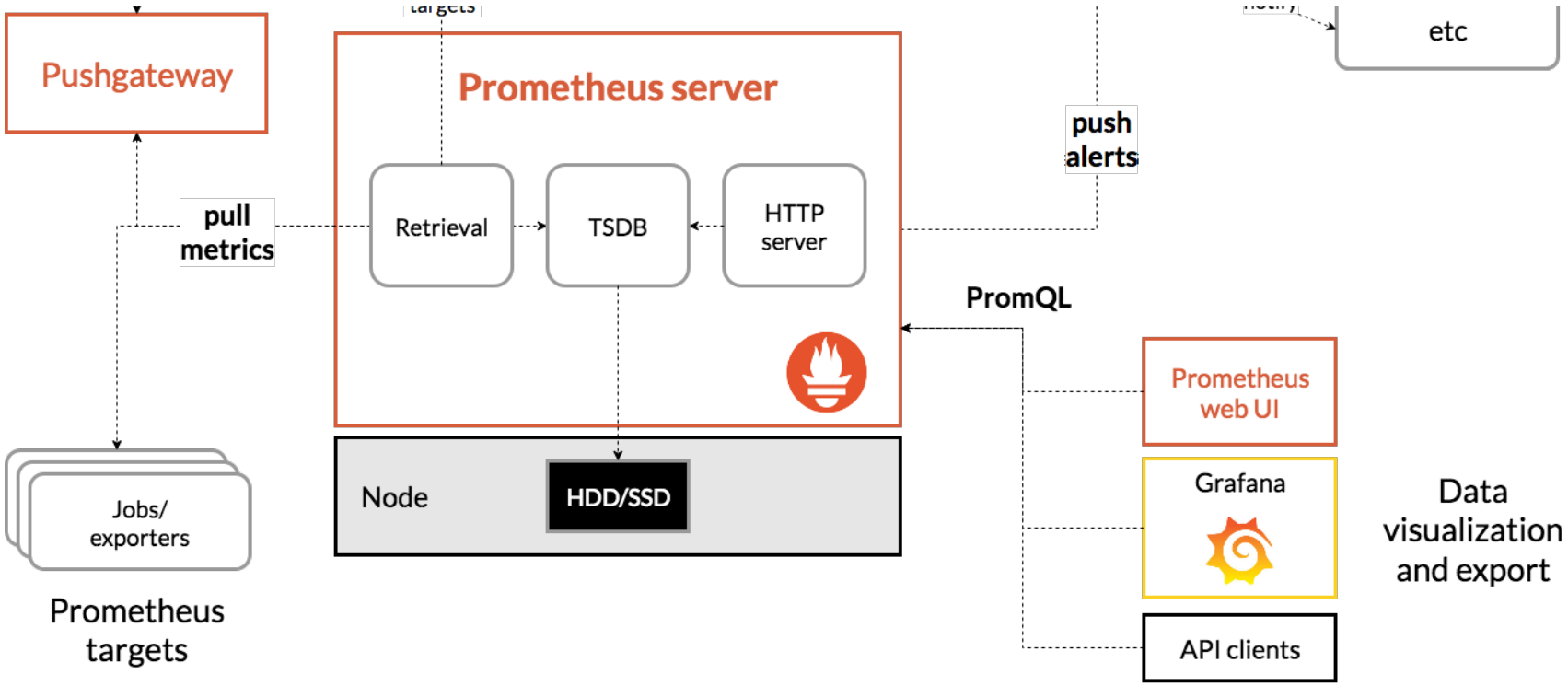
- An open-source system for automating deployment, scaling, and management of containerized applications



# Tools Introduction - Prometheus

- An open-source systems monitoring and alerting toolkit
- Collects and stores its metrics as time series data



# Tools Introduction - SQLite

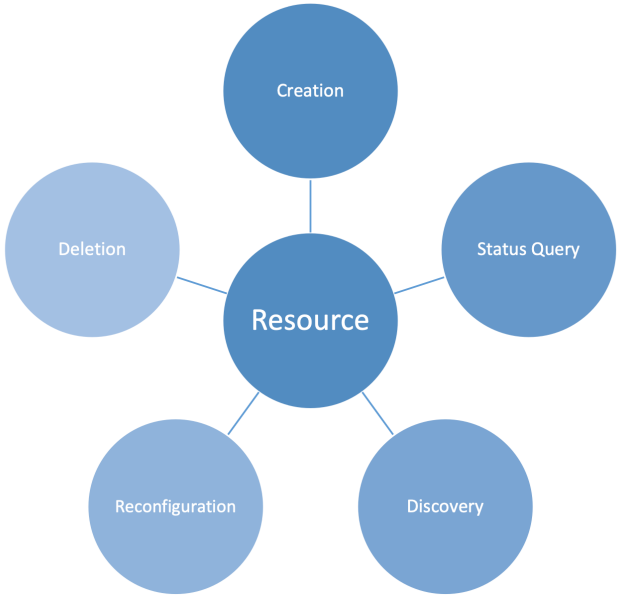
- A small and fast SQL database engine
- Single (cross-platform) file database
- Most widely deployed and used database engine
  - Every Android device
  - Every iPhone and iOS device
  - Every Mac
  - Every Windows10 machine
  - Every Firefox, Chrome, and Safari web browser

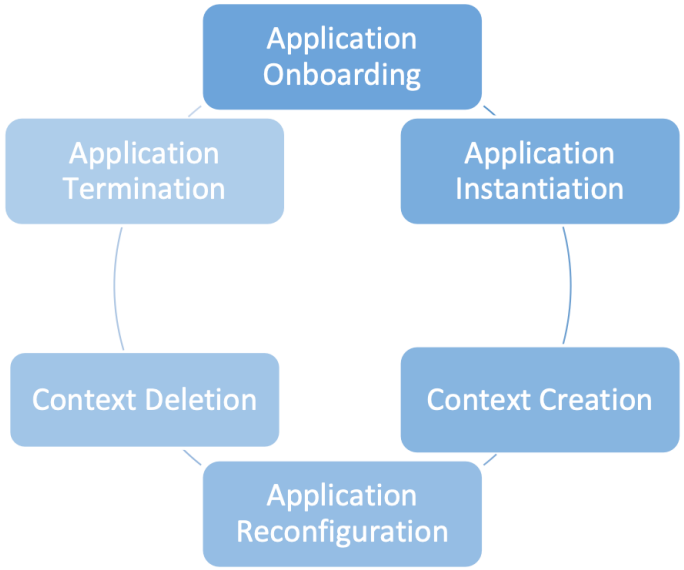
# Source Code Structure

- Python for main program
  - Flask for exposing API and website
  - SQLAlchemy as ORM to connect to SQLite database
  - Fabric to execute commands on remote server via ssh tunnel
  - Scapy to perform network packet transmission (arp, ping)
  - Kubernetes-client to communicate with kubernetes cluster
- Shell script to start the program in develop/production mode
- Shell script to install prerequisite on EFO, Control/Compute node

# P1935 API vs TestBed API

P1935 API



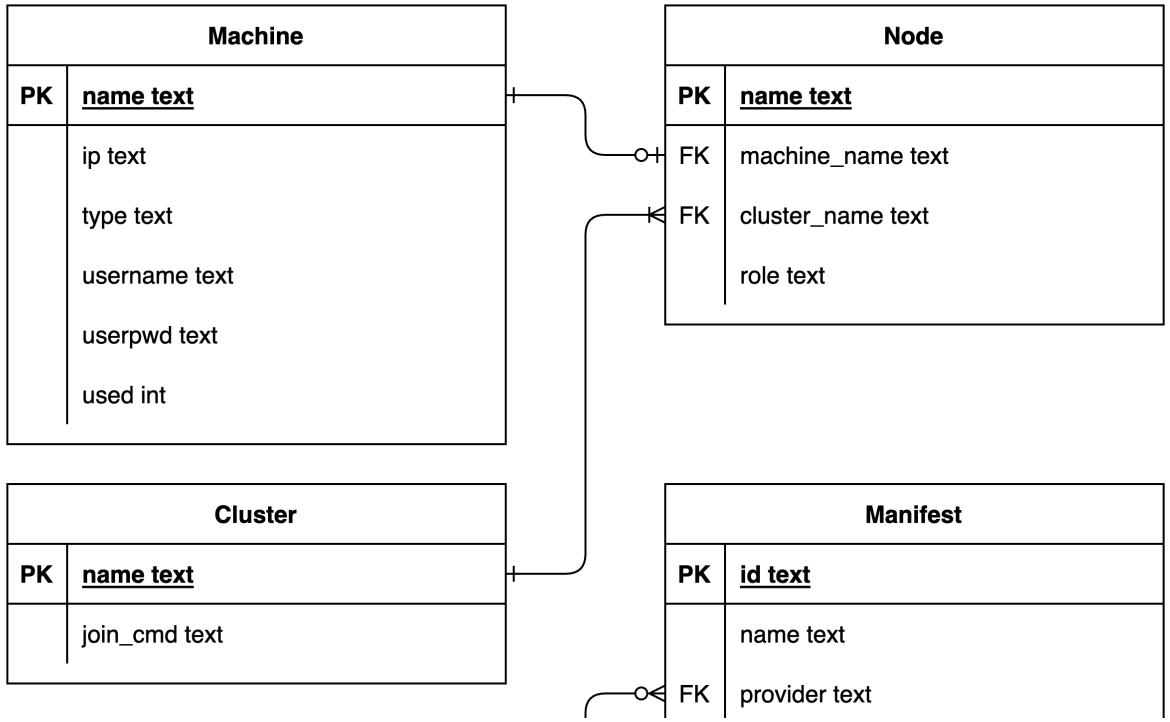




TestBed API

| Method | HTTP request | Description |
|---|---|---|
| list_manifests | `GET /api/manifests` | list all manifests |
| get_manifest | `GET /api/manifests/<manifestID>` | get a manifest |
| create_manifest | `POST /api/manifests` | onboard app |
| delete_manifest | `DELETE /api/manifests/<manifestID>` | delete a manifest |
| patch_manifest | `PATCH /api/manifests/<manifestID>` | reconfigure app |

# P1935 API - Resource Management

- In P1935 Standard implementation, resource can be categorized as
  - Machine
    - The server can be reached by EFO
  - Node
    - Control/Compute node
  - Cluster
    - Different Control node



# P1935 API - Resource Creation

- Create a resource
  1. Check the server is valid or create a VM on a server (create *Machine*)
  2. Select a machine as control node (create *Cluster*)
  3. Select a machine as compute node (create *Node*)

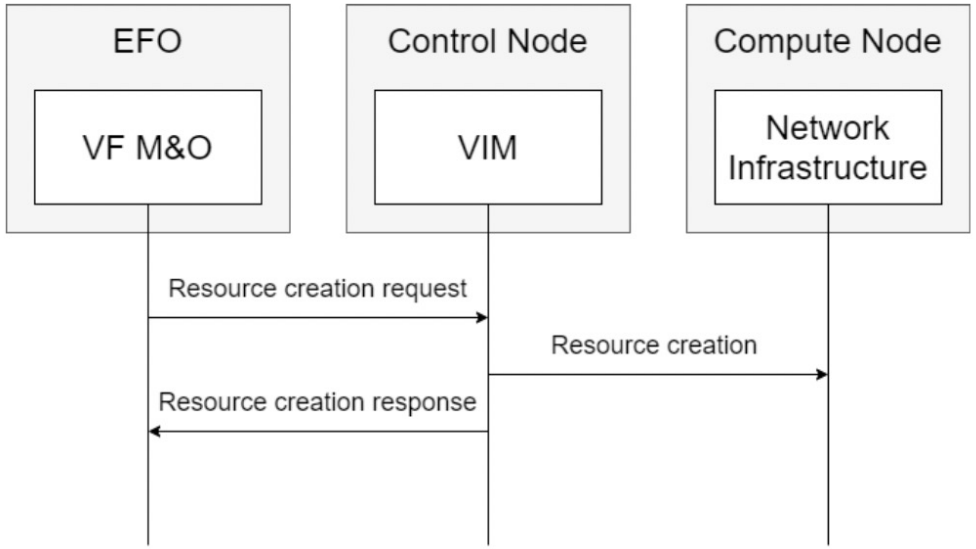



Figure 3.2.1.2-1 The workflow of the resource creation

# P1935 API - Application

- In P1935 Standard implementation, application can be categorized as
  - Manifest
    - Application blueprint (image url, cpu/memory limit)
    - In Yaml format
    - Need authorization
  - Application
    - Application runtime info (execution status, application url)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: helloworld
  labels:
    app: helloworld
  annotations:
    description: helloworld application
spec:
  replicas: 1
  selector:
    matchLabels:
      app: helloworld
  template:
    metadata:
      labels:
        app: helloworld
    spec:
      containers:
      - name: helloworld
        image: dingyiyi0226/demoapp
        ports:
        - containerPort: 5678
```

Manifest sample

# P1935 API – App Onboarding / Instantiation

- Application onboarding/instantiation
  1. Service provider upload manifest (create *Manifest*)
  2. Service operator authorize the manifest (update *Manifest*)
  3. Service operator instantiate the authorized manifest (create *Application*)

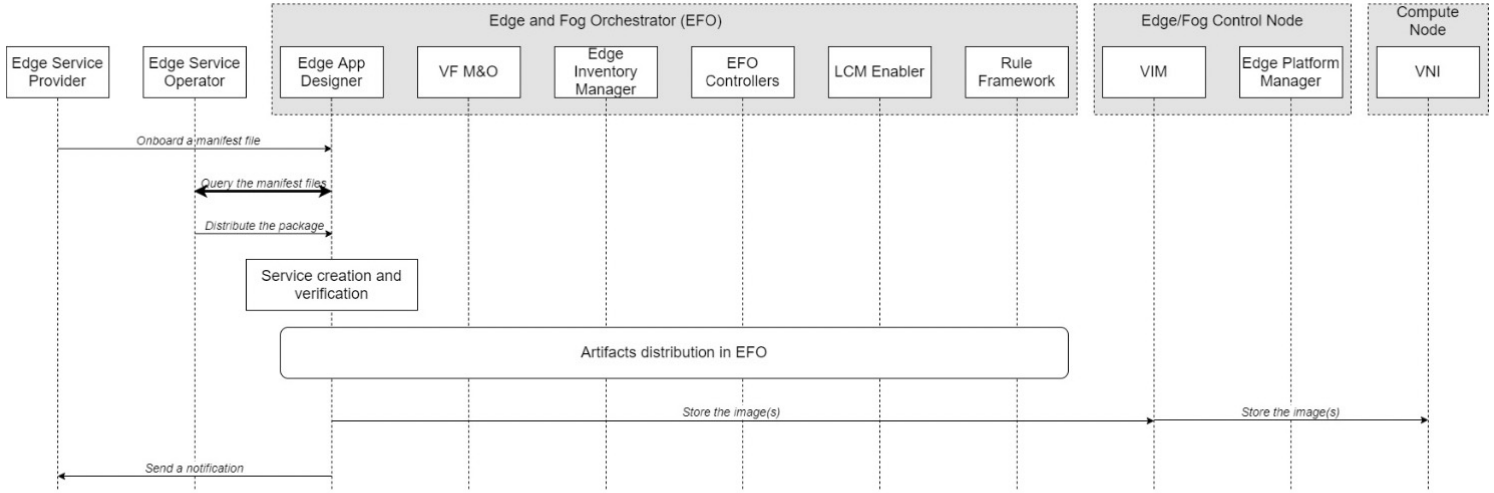



Figure 4.2.1.2-1 the workflow of application onboarding procedures

# Documentation

- A detailed documentation of testbed API and P1935 API

## Resource Creation

This operation API is used to create the target resource in the Edge/Fog system. It inclu
resources and virtual resources.

In the testbed implementation, the resources can be classified into three categories: ma

1. A new server can be managed by EFO after creating the machine from EFO
2. EFO can create a cluster from the machine list
3. EFO can create a node to a cluster from the machine list.

### API Requests

1. Infrastucture Owner create (join) a machine by create_machine
2. Infrastucture Owner create a cluster by create_cluster
3. Infrastucture Owner create a cluster by create_node
4. Infrastucture Owner can create a vm from specific IP by create_vm

P1935 API and testbed API relationship

## create_machine

Create a machine

### Request

```
POST /api/machines
```

**Request body**

| Property name | Value | Description |
|---|---|---|
| `ip` | `string` | machine ip address |
| `name` | `string` | machine name |
| `username` | `string` | username on this machine (must have sudo privilege) |
| `userpwd` | `string` | user password on this machine |
| `type` | `string` | machine type (Virtual, BS, EPC) |

### Response

If successful, this method returns `ip` in the response body

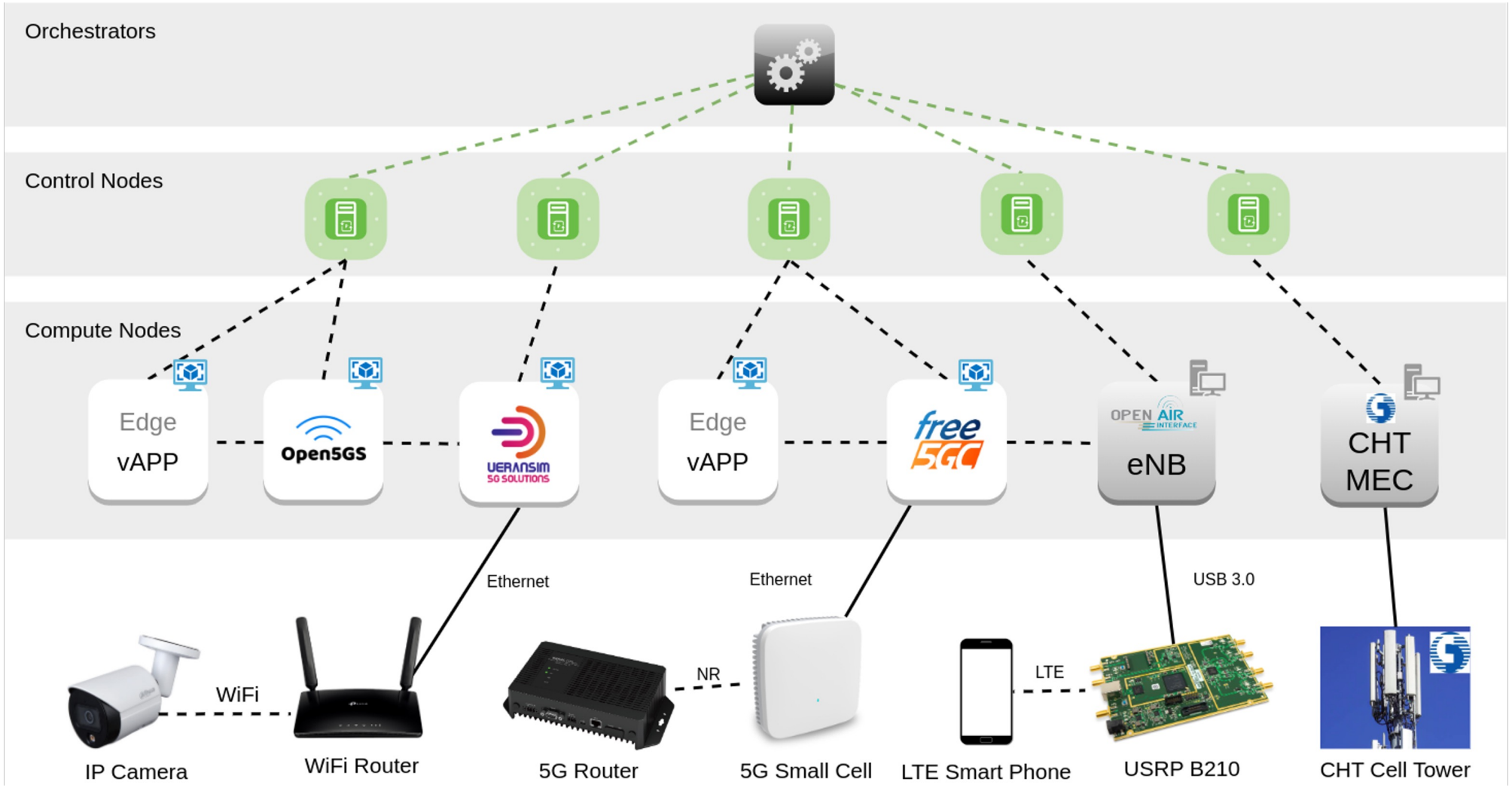
**Response details**

| Status | Description |
|---|---|
| **200** OK | |
| **400** Bad Request | Action failed |
| **401** Unauthorized | User is not authenticated |
| **403** Forbidden | User is not authorized |

# P1935 Network Configuration and DDoS Detection

Presenter: 黄旭弘

# P1935 5G RAN and Core Testbed



# Open5GS Installation

# Open5GS Installation

**Requirements**

1. Ubuntu 18.04 or later
2. CPU Cores: 2
3. RAM: 4G

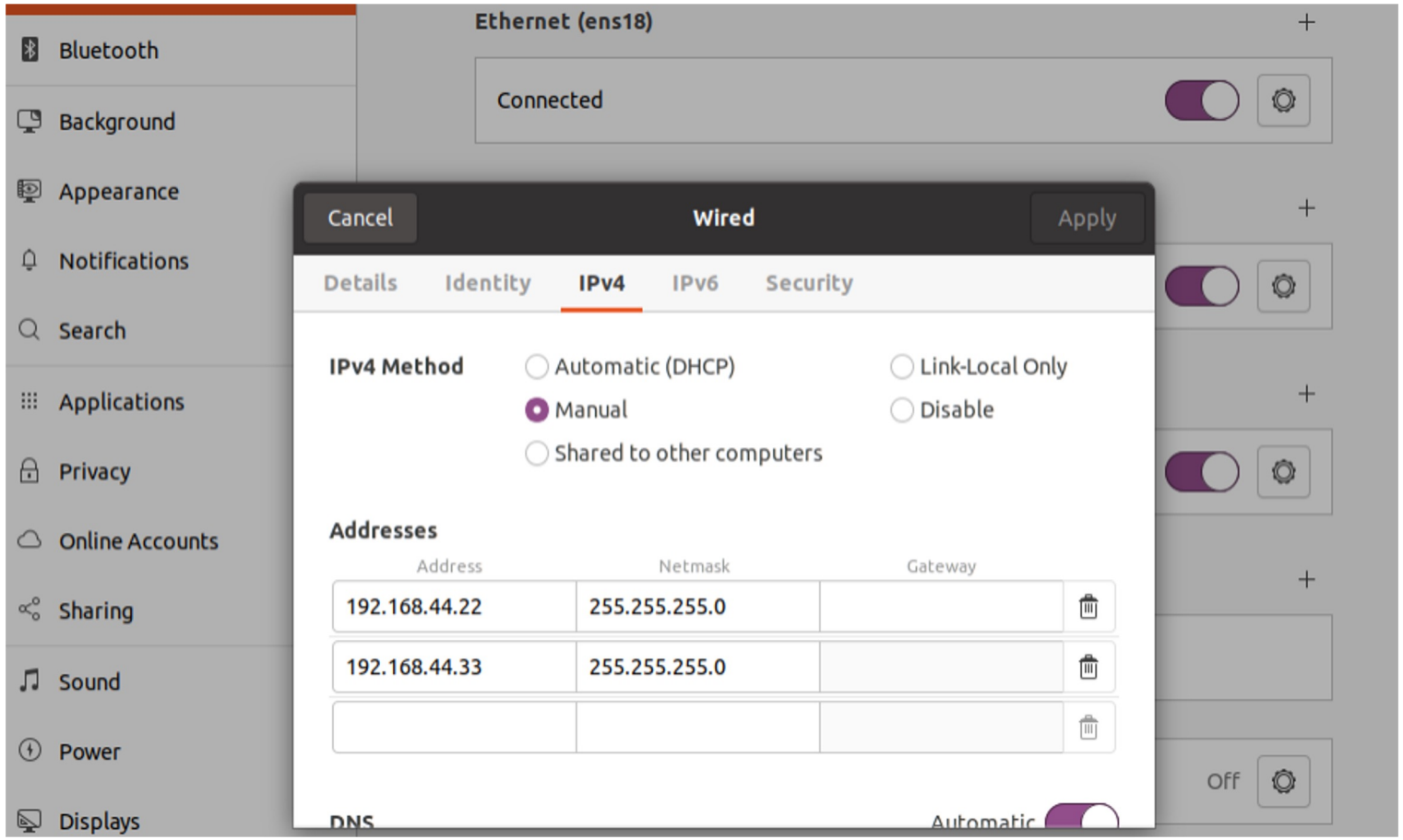
**Open5GS**

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository ppa:open5gs/latest
sudo apt update
sudo apt install open5gs
```

**WebGUI for Subscribe Editing**

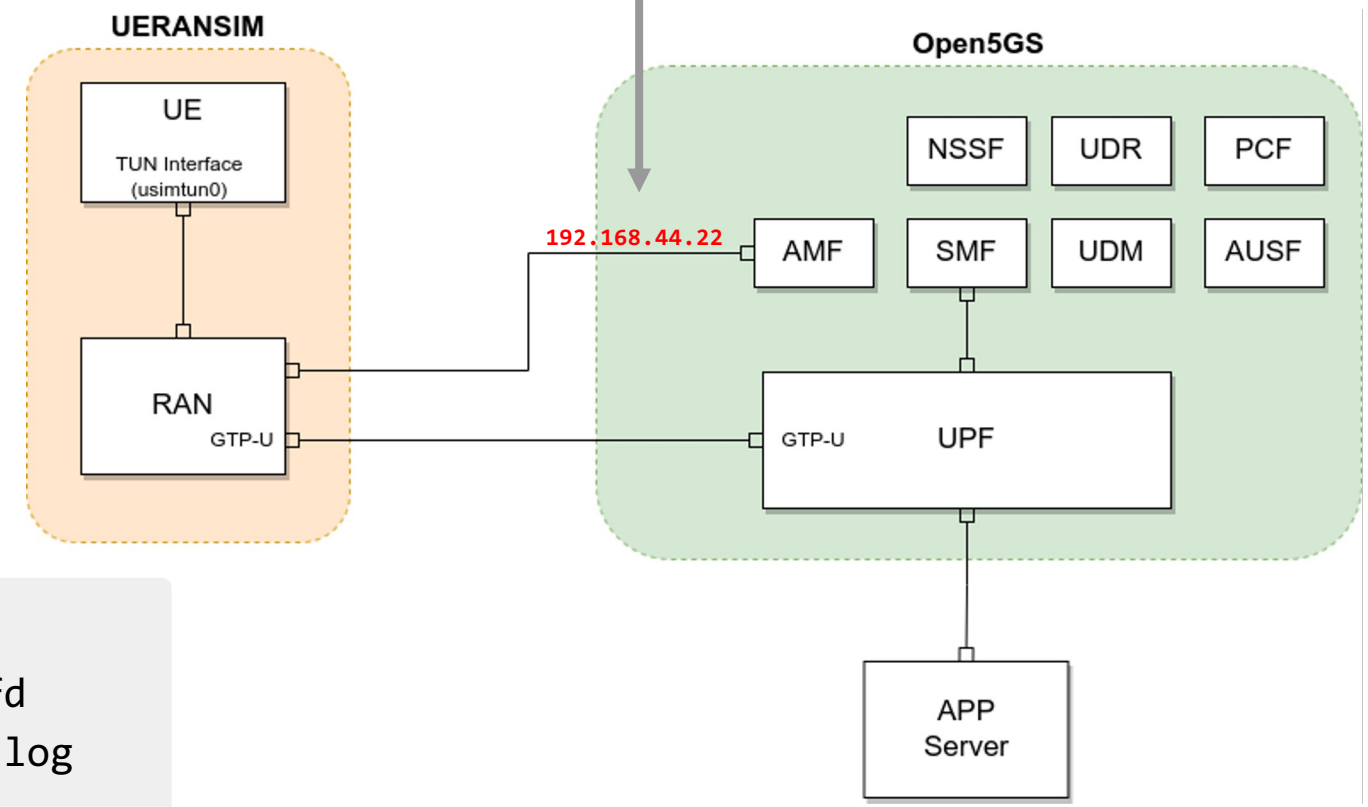
```
sudo apt update
sudo apt install curl
curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash
-
sudo apt install nodejs
cd ~; git clone https://github.com/open5gs/open5gs.git
```

# Setup IP for N2/N3 Interface



# Setup Open5GS AMF

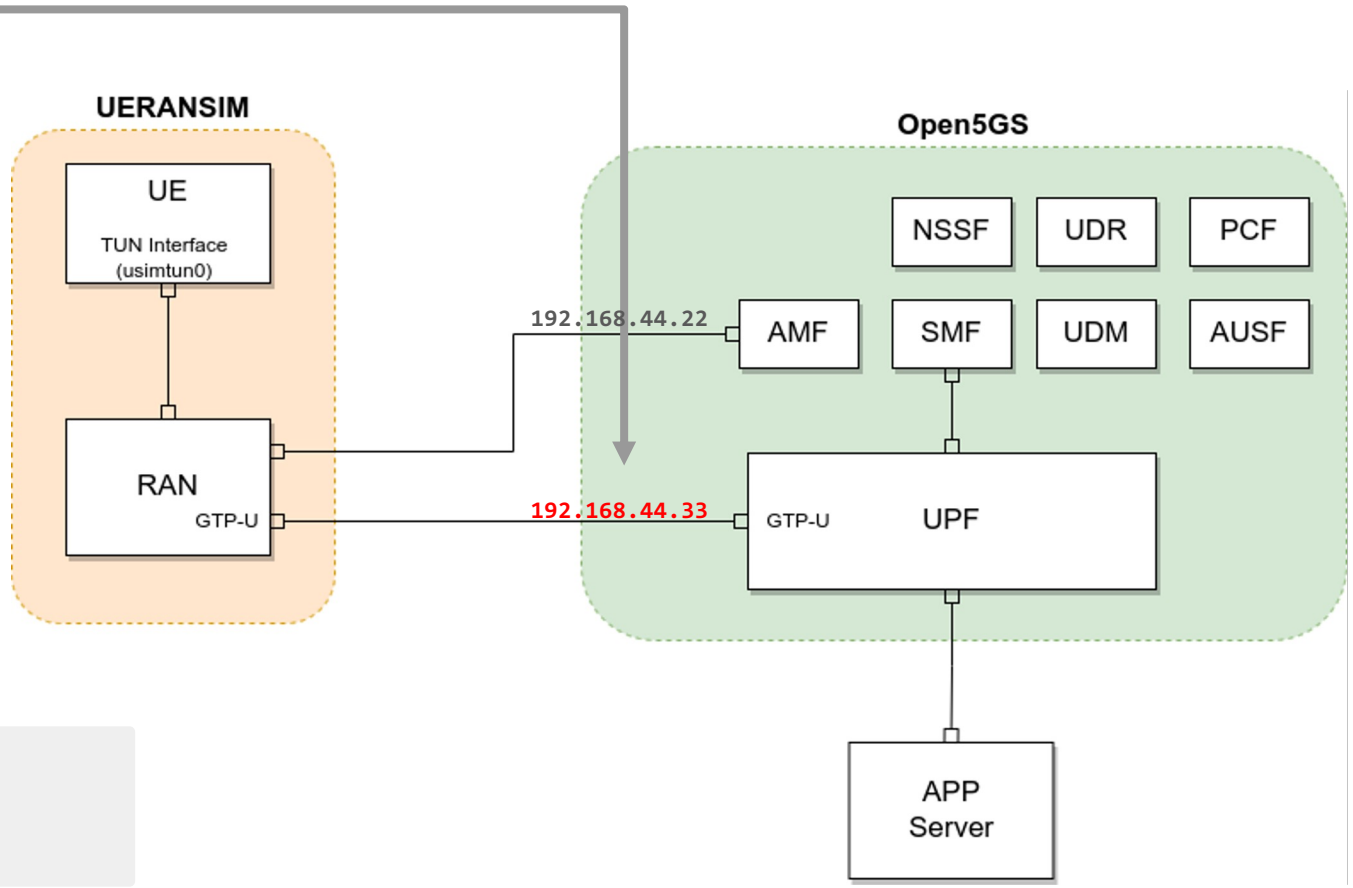
```
# /etc/amf.cfg
amf:
        sbi:
        - addr: 127.0.0.5
        port: 7777
        ngap:
        - addr: 192.168.44.22
```




```
sudo systemctl restart open5gs-amfd
sudo tail -f /var/log/open5gs/amf.log
```

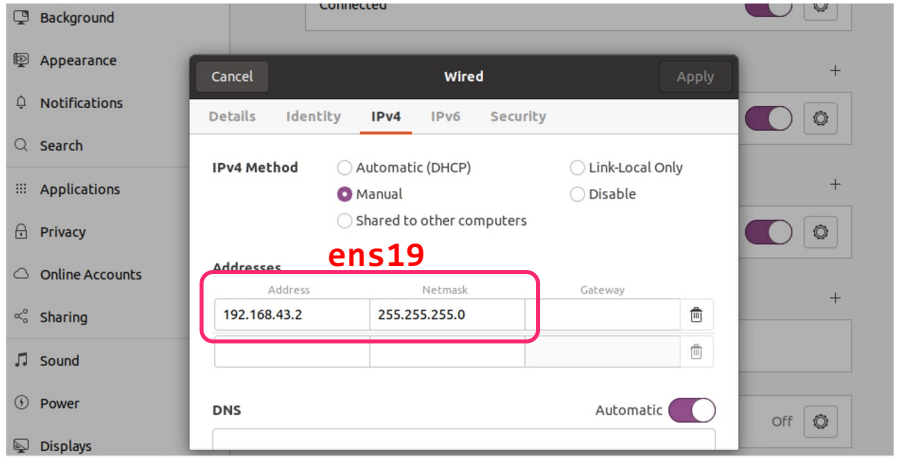
# Setup Open5GS UPF

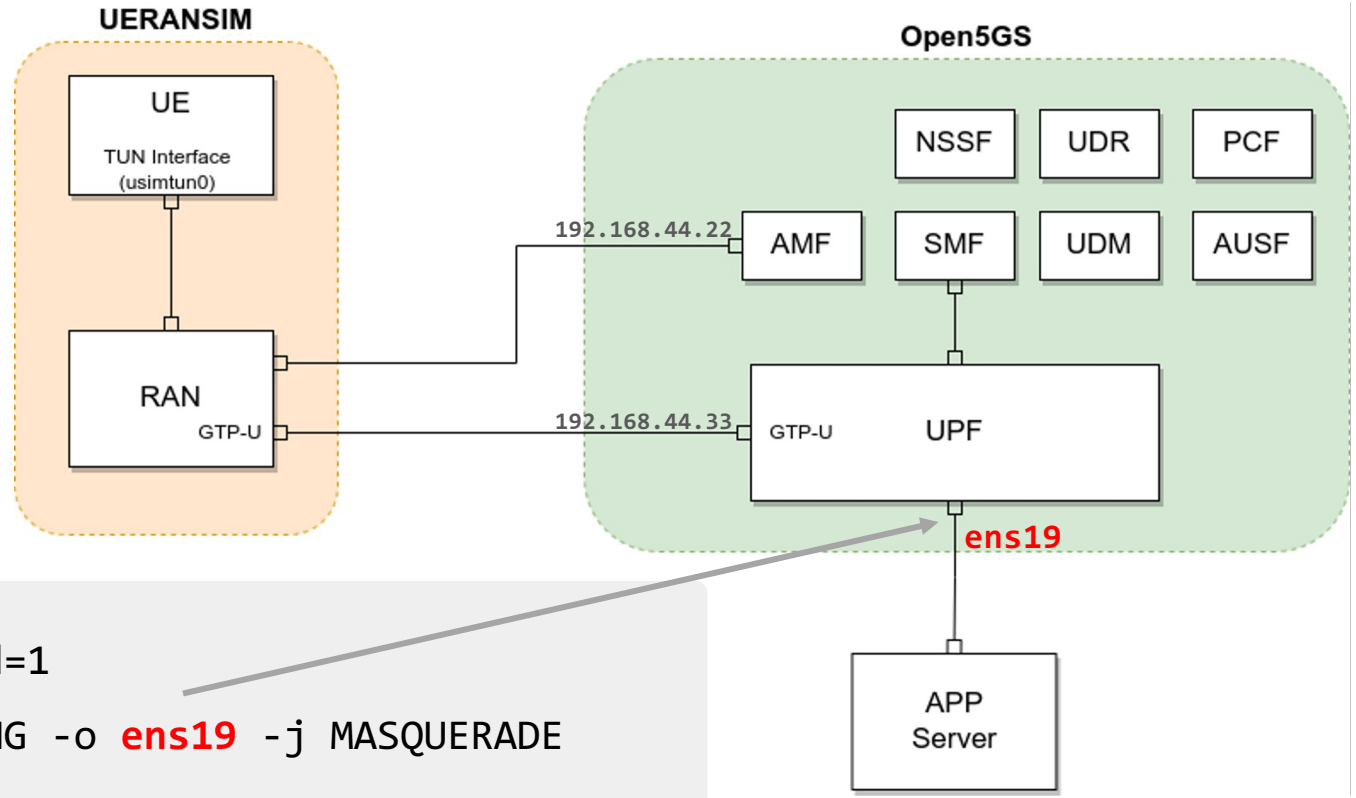
```
# /etc/upf.cfg
upf:
        pfcp:
        - addr: 127.0.0.7
        gtpu:
        - addr: 192.168.44.33
        subnet:
        - addr: 10.45.0.1/16
        - addr: 2001:db8:cafe::1/48
```




```
sudo systemctl restart open5gs-upfd
```

# Setup IP and NAT for N6 Interface



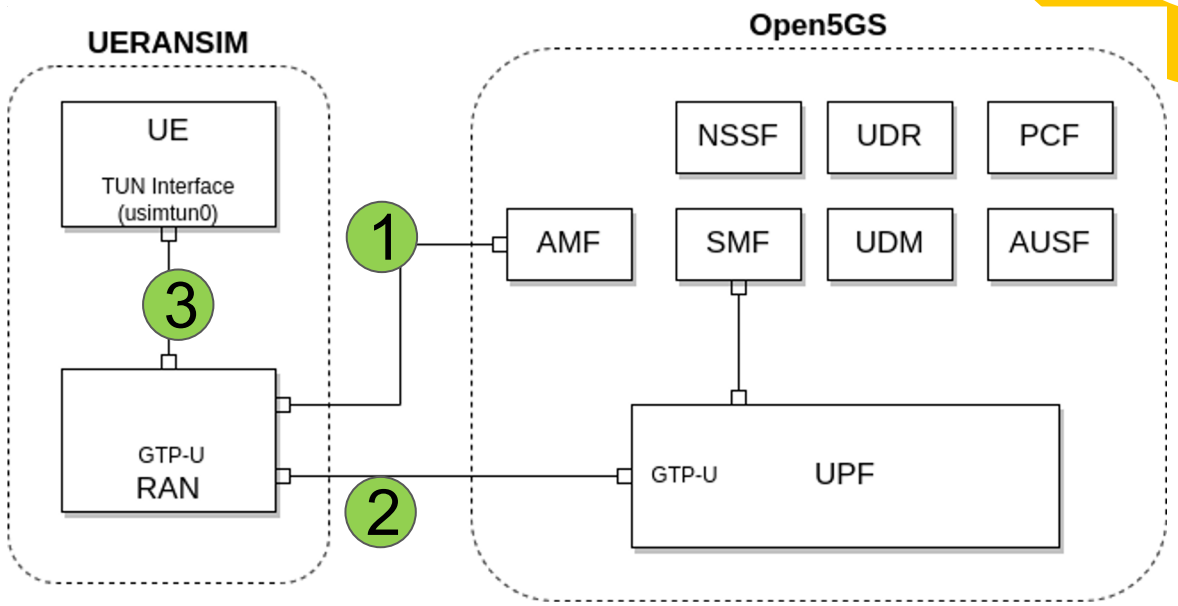

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o ens19 -j MASQUERADE
sudo systemctl stop ufw
sudo iptables -I FORWARD 1 -j ACCEPT
```

# UERANSIM Installation

# What is UERANSIM

## Interface in UE/RAN:

1. Control Interface (between RAN and AMF)
2. User Interface (between RAN and UPF)
3. Radio Interface (between UE and RAN)




## Control Plane:

A. NAS is in control of UE

B. NGAP is in control of RAN.

## Limitation:

- UERANSIM does not implement 5G radio protocols below the RRC layer. PHY, MAC, RLC, PDCP is not implemented in UERANSIM.

# UERANSIM Installation

## Requirements

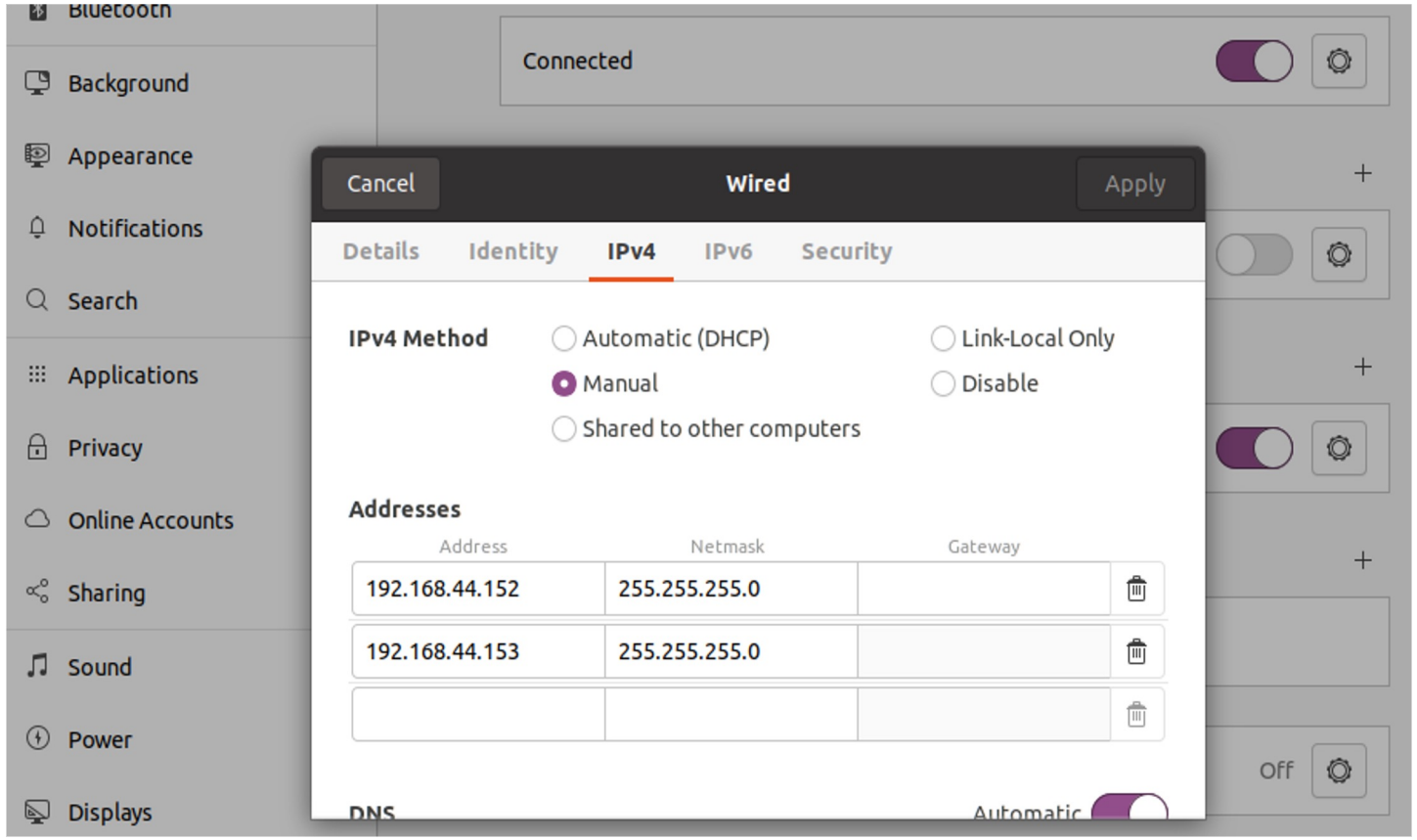
1. Ubuntu 16.04 or later
2. CPU Cores: 4
3. RAM: 4G
4. **CMake 3.17 or later**
5. gcc 9.0.0 or later
6. g++ 9.0.0 or later

```
sudo apt update
sudo apt upgrade
sudo apt install make g++ libsctp-dev lksctp-tools iproute2
sudo apt remove cmake
sudo snap install cmake --classic

git clone https://github.com/aligungr/UERANSIM
cd UERANSIM
make
```

# UERANSIM Throughput CPU/Memory Usage

```
1  [||||||||||||||||||||                                    36.7%]   Tasks: 142, 323 thr; 4 running
2  [||||||||||||||||||||||||||||||||||||||||||||||||||||||   73.1%]   Load average: 1.84 1.40 0.63
3  [||||||||||||||||||||                                    34.7%]   Uptime: 1 day, 20:48:12
4  [||||||||||||||                                          25.9%]
Mem[||||||||||||||||||||||                                997M/7.76G]
Swp[                                                       0K/2.00G]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
42081 compal     20   0  738M  9004  5808 S 84.4  0.1  3:20.90 ./build/nr-gnb -c config/open5gs-pve04-gnb.yaml
42094 root       20   0  601M  5972  4320 S 87.8  0.1  3:23.90 ./build/nr-ue -c config/open5gs-ue.yaml
42091 compal     20   0  738M  9004  5808 R 39.8  0.1  1:27.06 ./build/nr-gnb -c config/open5gs-pve04-gnb.yaml
42100 root       20   0  601M  5972  4320 S 29.5  0.1  1:08.90 ./build/nr-ue -c config/open5gs-ue.yaml
42087 compal     20   0  738M  9004  5808 S 22.6  0.1  0:54.16 ./build/nr-gnb -c config/open5gs-pve04-gnb.yaml
42116 root       20   0  601M  5972  4320 S 15.8  0.1  0:34.13 ./build/nr-ue -c config/open5gs-ue.yaml
42088 compal     20   0  738M  9004  5808 S 13.0  0.1  0:32.97 ./build/nr-gnb -c config/open5gs-pve04-gnb.yaml
42101 root       20   0  601M  5972  4320 S 11.7  0.1  0:26.28 ./build/nr-ue -c config/open5gs-ue.yaml
42089 compal     20   0  738M  9004  5808 S  9.6  0.1  0:26.61 ./build/nr-gnb -c config/open5gs-pve04-gnb.yaml
42117 root       20   0  601M  5972  4320 R 10.3  0.1  0:25.37 ./build/nr-ue -c config/open5gs-ue.yaml
42102 root       20   0  601M  5972  4320 S  9.6  0.1  0:25.26 ./build/nr-ue -c config/open5gs-ue.yaml
42097 root       20   0  601M  5972  4320 S  9.6  0.1  0:23.81 ./build/nr-ue -c config/open5gs-ue.yaml
 1597 compal     20   0 4632M  355M  130M S  1.4  4.5  4:46.90 /usr/bin/gnome-shell
 1616 compal     20   0 4632M  355M  130M S  0.7  4.5  0:34.55 /usr/bin/gnome-shell
 1614 compal     20   0 4632M  355M  130M S  0.0  4.5  0:31.95 /usr/bin/gnome-shell
 1440 compal     20   0  803M 93540 51680 S  0.0  1.1  0:26.25 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/X
42888 compal     20   0 11548  4476  3288 R  0.7  0.1  0:01.03 htop
 1615 compal     20   0 4632M  355M  130M S  0.0  4.5  0:31.24 /usr/bin/gnome-shell
 1613 compal     20   0 4632M  355M  130M S  0.0  4.5  0:32.62 /usr/bin/gnome-shell
27881 compal     20   0  803M 86044 36952 S  1.4  1.1  2:10.23 /home/compal/.vscode-server/bin/e4503b30fc78200f846c62cf8091b7
 1916 compal     20   0  891M 73012 46272 S  0.0  0.9  0:04.90 /usr/bin/python3 /usr/bin/terminator
10083 compal     20   0  921M 85064 36104 S  0.0  1.0  1:19.29 /home/compal/.vscode-server/bin/e4503b30fc78200f846c62cf8091b7
```

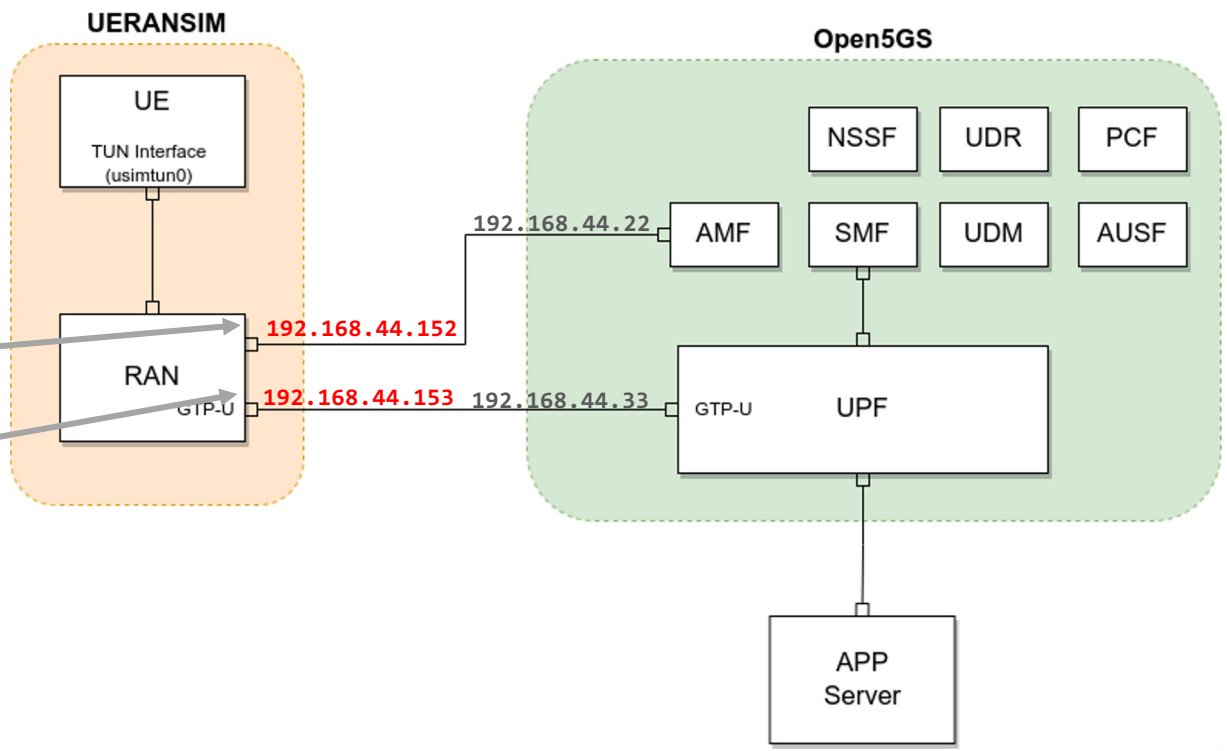
# UERANSIM N2/N3 Interface IP



# UERANSIM gNB Configuration

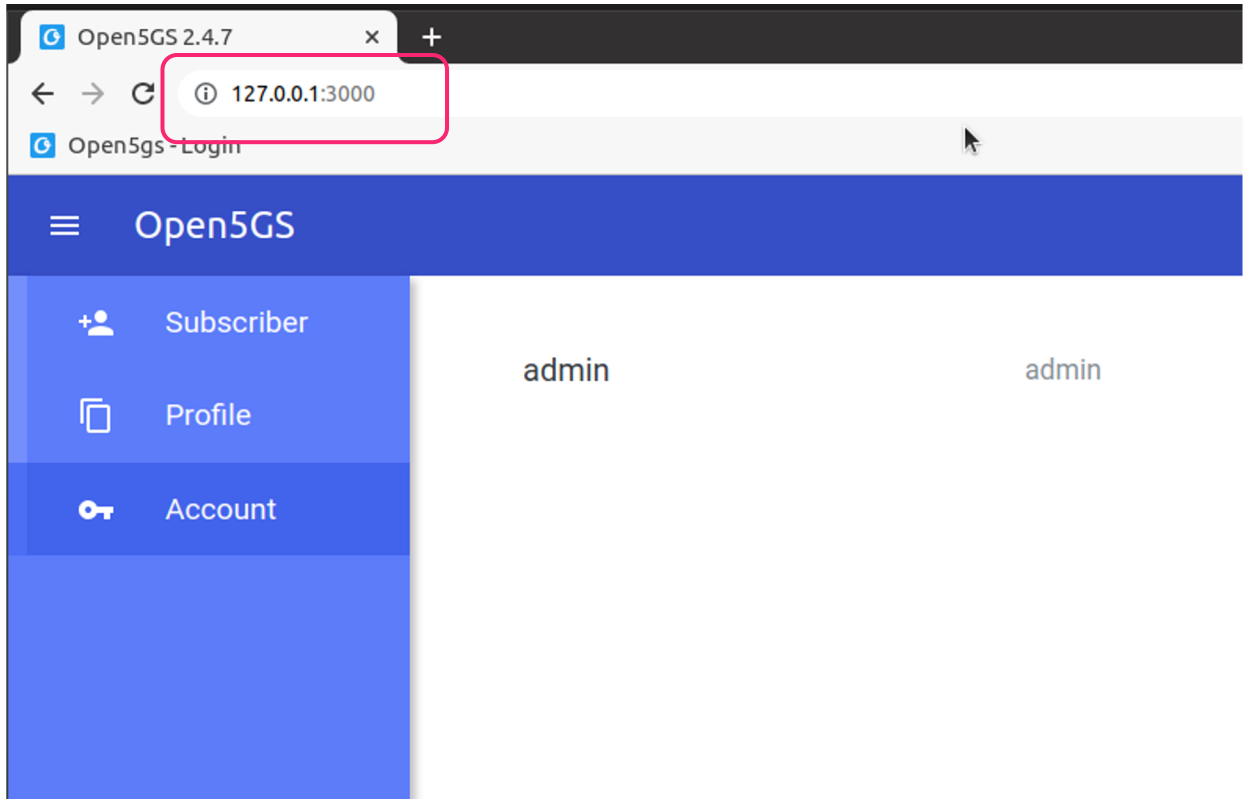
~/UERANSIM/config/open5gs-gnb.yaml

```yaml
mcc: '999'   # Mobile Country Code value
mnc: '70'    # Mobile Network Code value
nci: '0x000000010'  # NR Cell Identity (36-bit)
idLength: 32        # NR gNB ID length in bits
tac: 1              # NTracking Area Code

# gNB's local IP address for Radio Link
linkIp: 127.0.0.1

# gNB's local IP address for N2 Interface
ngapIp: 192.168.44.152

# gNB's local IP address for N3 Interface
gtpIp: 192.168.44.153

# List of AMF address information
amfConfigs:
  - address: 192.168.44.22
          port: 38412
```



# Start Open5GS WebGUI Server

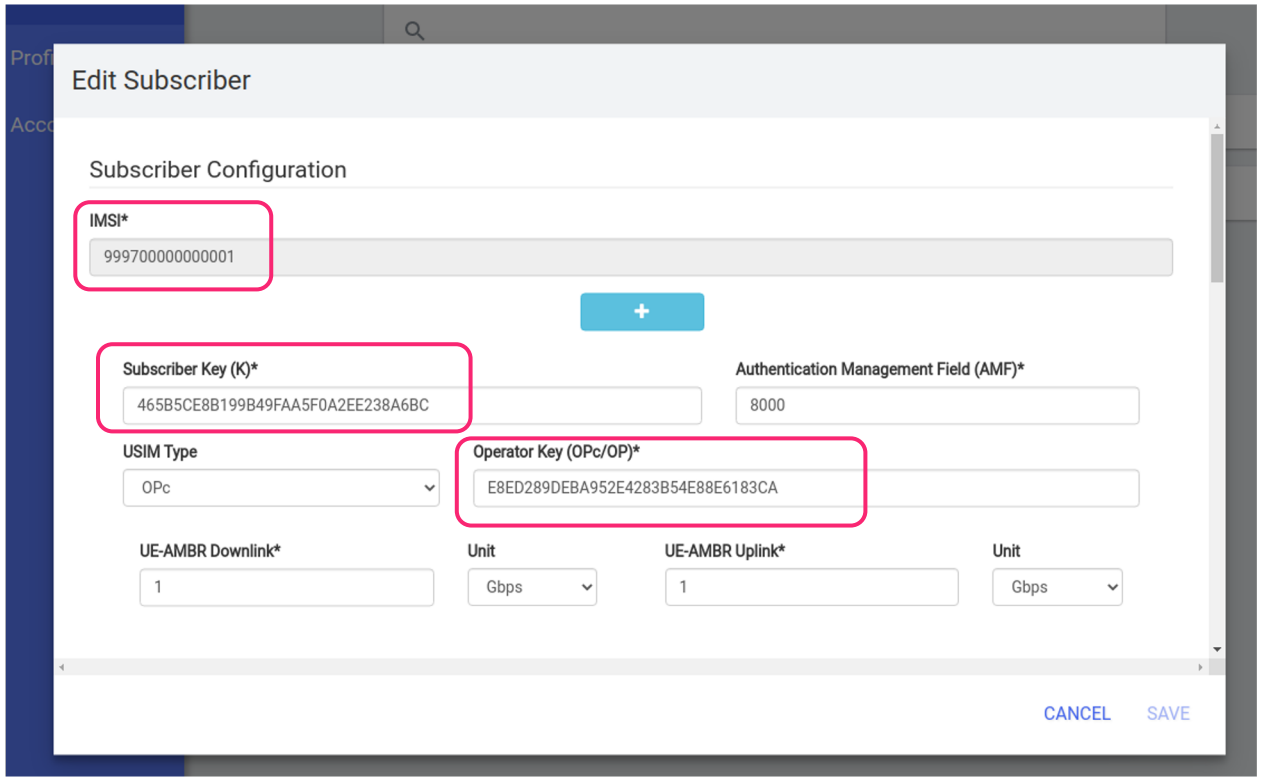
```
# run webui with npm
cd ~/open5gs/webui
npm i
npm audit fix
npm run dev --host 0.0.0.0

# the web interface will start on
http://127.0.0.1:3000

# webui login credentials
username - admin
password - 1423
```



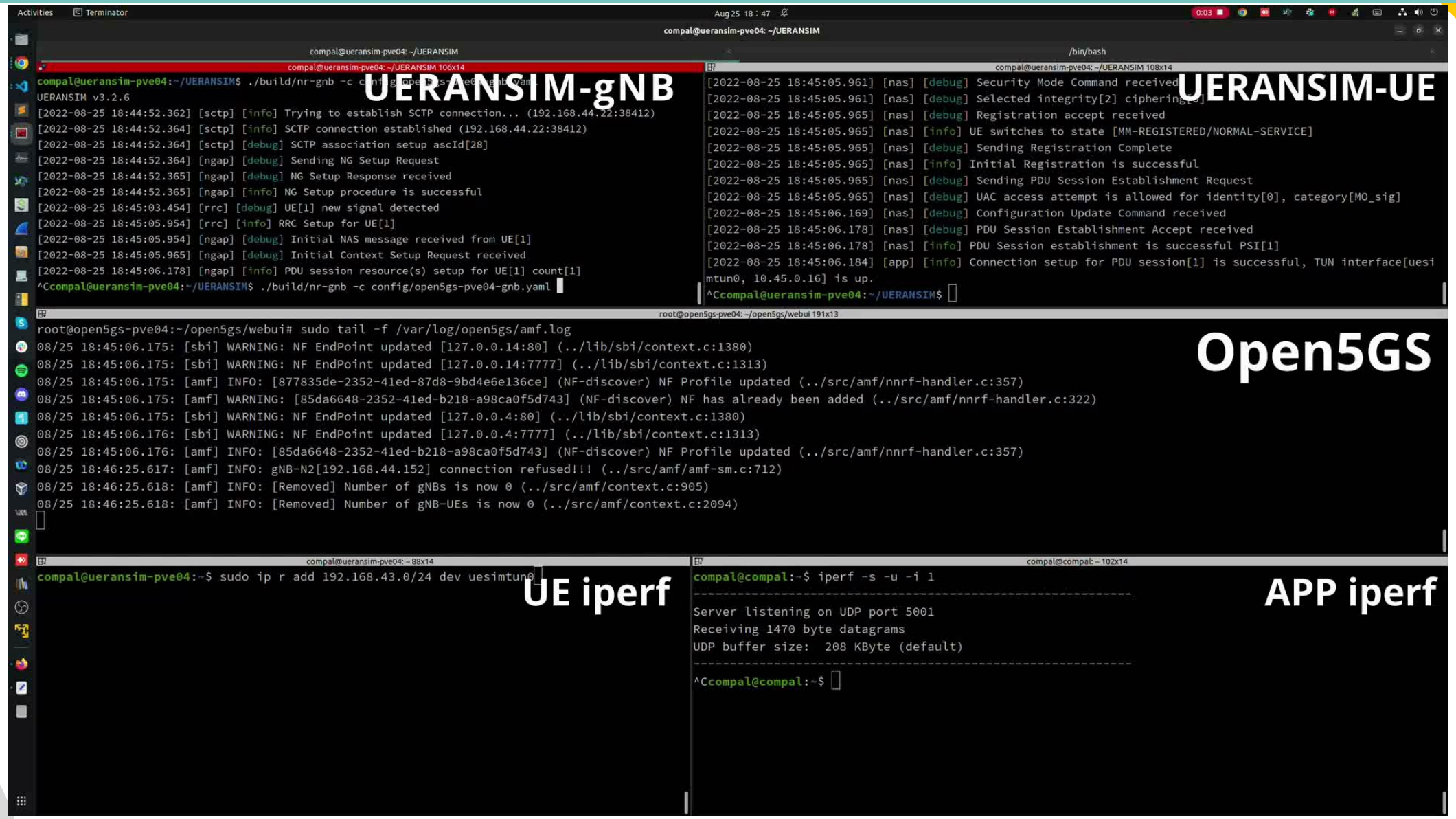
# Add Subscriber by WebGUI

~/UERANSIM/config/open5gs-ue.yaml

```yaml
# Add SIM info
supi: 'imsi-999700000000001'

# Mobile Country Code value of HPLMN
mcc: '999'

# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '70'

# Permanent subscription key
key: '465B5CE8B199B49FAA5F0A2EE238A6BC'

# Operator code (OP or OPC) of the UE
op: 'E8ED289DEBA952E4283B54E88E6183CA'

opType: 'OPC'
```



# Demo


UERANSIM-gNB
compal@ueransim-pve04:~/UERANSIM$ ./build/nr-gnb -c
UERANSIM v3.2.6
[2022-08-25 18:44:52.362] [sctp] [info] Trying to establish SCTP connection... (192.168.44.22:38412)
[2022-08-25 18:44:52.364] [sctp] [info] SCTP connection established (192.168.44.22:38412)
[2022-08-25 18:44:52.364] [sctp] [debug] SCTP association setup ascId[28]
[2022-08-25 18:44:52.364] [ngap] [debug] Sending NG Setup Request
[2022-08-25 18:44:52.365] [ngap] [debug] NG Setup Response received
[2022-08-25 18:44:52.365] [ngap] [info] NG Setup procedure is successful
[2022-08-25 18:45:03.454] [rrc] [debug] UE[1] new signal detected
[2022-08-25 18:45:05.954] [rrc] [info] RRC Setup for UE[1]
[2022-08-25 18:45:05.954] [ngap] [debug] Initial NAS message received from UE[1]
[2022-08-25 18:45:05.965] [ngap] [debug] Initial Context Setup Request received
[2022-08-25 18:45:06.178] [ngap] [info] PDU session resource(s) setup for UE[1] count[1]
^Ccompal@ueransim-pve04:~/UERANSIM$ ./build/nr-gnb -c config/open5gs-pve04-gnb.yaml
UERANSIM-UE
[2022-08-25 18:45:05.961] [nas] [debug] Security Mode Command received
[2022-08-25 18:45:05.961] [nas] [debug] Selected integrity[2] ciphering[0]
[2022-08-25 18:45:05.965] [nas] [debug] Registration accept received
[2022-08-25 18:45:05.965] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
[2022-08-25 18:45:05.965] [nas] [debug] Sending Registration Complete
[2022-08-25 18:45:05.965] [nas] [info] Initial Registration is successful
[2022-08-25 18:45:05.965] [nas] [debug] Sending PDU Session Establishment Request
[2022-08-25 18:45:05.965] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2022-08-25 18:45:06.169] [nas] [debug] Configuration Update Command received
[2022-08-25 18:45:06.178] [nas] [debug] PDU Session Establishment Accept received
[2022-08-25 18:45:06.178] [nas] [info] PDU Session establishment is successful PSI[1]
[2022-08-25 18:45:06.184] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.45.0.16] is up.
^Ccompal@ueransim-pve04:~/UERANSIM$
Open5GS
root@open5gs-pve04:~/open5gs/webui# sudo tail -f /var/log/open5gs/amf.log
08/25 18:45:06.175: [sbi] WARNING: NF EndPoint updated [127.0.0.14:80] (../lib/sbi/context.c:1380)
08/25 18:45:06.175: [sbi] WARNING: NF EndPoint updated [127.0.0.14:7777] (../lib/sbi/context.c:1313)
08/25 18:45:06.175: [amf] INFO: [877835de-2352-41ed-87d8-9bd4e6e136ce] (NF-discover) NF Profile updated (../src/amf/nnrf-handler.c:357)
08/25 18:45:06.175: [amf] WARNING: [85da6648-2352-41ed-b218-a98ca0f5d743] (NF-discover) NF has already been added (../src/amf/nnrf-handler.c:322)
08/25 18:45:06.175: [sbi] WARNING: NF EndPoint updated [127.0.0.4:80] (../lib/sbi/context.c:1380)
08/25 18:45:06.176: [sbi] WARNING: NF EndPoint updated [127.0.0.4:7777] (../lib/sbi/context.c:1313)
08/25 18:45:06.176: [amf] INFO: [85da6648-2352-41ed-b218-a98ca0f5d743] (NF-discover) NF Profile updated (../src/amf/nnrf-handler.c:357)
08/25 18:46:25.617: [amf] INFO: gNB-N2[192.168.44.152] connection refused!!! (../src/amf/amf-sm.c:712)
08/25 18:46:25.618: [amf] INFO: [Removed] Number of gNBs is now 0 (../src/amf/context.c:905)
08/25 18:46:25.618: [amf] INFO: [Removed] Number of gNB-UEs is now 0 (../src/amf/context.c:2094)
UE iperf
compal@ueransim-pve04:~$ sudo ip r add 192.168.43.0/24 dev uesimtun0
APP iperf
compal@compal:~$ iperf -s -u -i 1
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
^Ccompal@compal:~$

# Wireshark Packages

| | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000000 | 192.168.44.152 | 192.168.44.22 | SCTP | 82 | INIT |
| 2 | 0.000086363 | 192.168.44.22 | 192.168.44.152 | SCTP | 306 | INIT_ACK |
| 3 | 0.000171018 | 192.168.44.152 | 192.168.44.22 | SCTP | 278 | COOKIE_ECHO |
| 4 | 0.000206663 | 192.168.44.22 | 192.168.44.152 | SCTP | 50 | COOKIE_ACK |
| 5 | 0.000553105 | 192.168.44.152 | 192.168.44.22 | NGAP | 134 | NGSetupRequest |
| 6 | 0.000568845 | 192.168.44.22 | 192.168.44.152 | SCTP | 62 | SACK |
| 7 | 0.000661027 | 192.168.44.22 | 192.168.44.152 | NGAP | 118 | NGSetupResponse |
| 8 | 0.000763641 | 192.168.44.152 | 192.168.44.22 | SCTP | 62 | SACK |
| 9 | 5.644185476 | 192.168.44.152 | 192.168.44.22 | NGAP/NAS-5GS | 138 | InitialUEMessage, Registration request |
| 10 | 5.648075783 | 192.168.44.22 | 192.168.44.152 | NGAP/NAS-5GS | 146 | DownlinkNASTransport, Authentication request |
| 11 | 5.648758319 | 192.168.44.152 | 192.168.44.22 | NGAP/NAS-5GS | 146 | UplinkNASTransport, Authentication response |
| 12 | 5.650833073 | 192.168.44.22 | 192.168.44.152 | NGAP/NAS-5GS | 126 | DownlinkNASTransport |
| 13 | 5.651242513 | 192.168.44.152 | 192.168.44.22 | NGAP/NAS-5GS | 186 | UplinkNASTransport |
| 14 | 5.655747454 | 192.168.44.22 | 192.168.44.152 | NGAP/NAS-5GS | 230 | InitialContextSetupRequest |
| 15 | 5.656031337 | 192.168.44.152 | 192.168.44.22 | NGAP | 98 | InitialContextSetupResponse |
| 16 | 5.857287869 | 192.168.44.22 | 192.168.44.152 | SCTP | 62 | SACK |
| 17 | 5.857460393 | 192.168.44.152 | 192.168.44.22 | NGAP/NAS-5GS | 238 | UplinkNASTransport |
| 18 | 5.857669220 | 192.168.44.22 | 192.168.44.152 | NGAP/NAS-5GS | 142 | DownlinkNASTransport |
| 19 | 5.861124736 | 192.168.44.22 | 192.168.44.152 | NGAP/NAS-5GS | 242 | PDUSessionResourceSetupRequest |
| 20 | 5.861219493 | 192.168.44.152 | 192.168.44.22 | SCTP | 62 | SACK |
| 21 | 5.863797982 | 192.168.44.152 | 192.168.44.22 | NGAP | 102 | PDUSessionResourceSetupResponse |
| 22 | 6.065290456 | 192.168.44.22 | 192.168.44.152 | SCTP | 62 | SACK |
| 23 | 11.365306108 | 192.168.44.22 | 192.168.44.152 | SCTP | 106 | HEARTBEAT |
| 24 | 11.365498606 | 192.168.44.152 | 192.168.44.22 | SCTP | 106 | HEARTBEAT_ACK |
| 25 | 17.249366519 | 192.168.44.22 | 192.168.44.152 | SCTP | 106 | HEARTBEAT |
| 26 | 17.249549747 | 192.168.44.152 | 192.168.44.22 | SCTP | 106 | HEARTBEAT_ACK |
| 27 | 23.649308355 | 192.168.44.22 | 192.168.44.152 | SCTP | 106 | HEARTBEAT |
| 28 | 23.649538190 | 192.168.44.152 | 192.168.44.22 | SCTP | 106 | HEARTBEAT_ACK |
| 29 | 28.673644842 | 10.45.0.19 | 192.168.43.3 | GTP <ICMP> | 142 | Echo (ping) request id=0x0029, seq=1/256, ttl=64 (r |
| 30 | 28.674099395 | 192.168.43.3 | 10.45.0.19 | GTP <ICMP> | 142 | Echo (ping) reply id=0x0029, seq=1/256, ttl=63 (r |

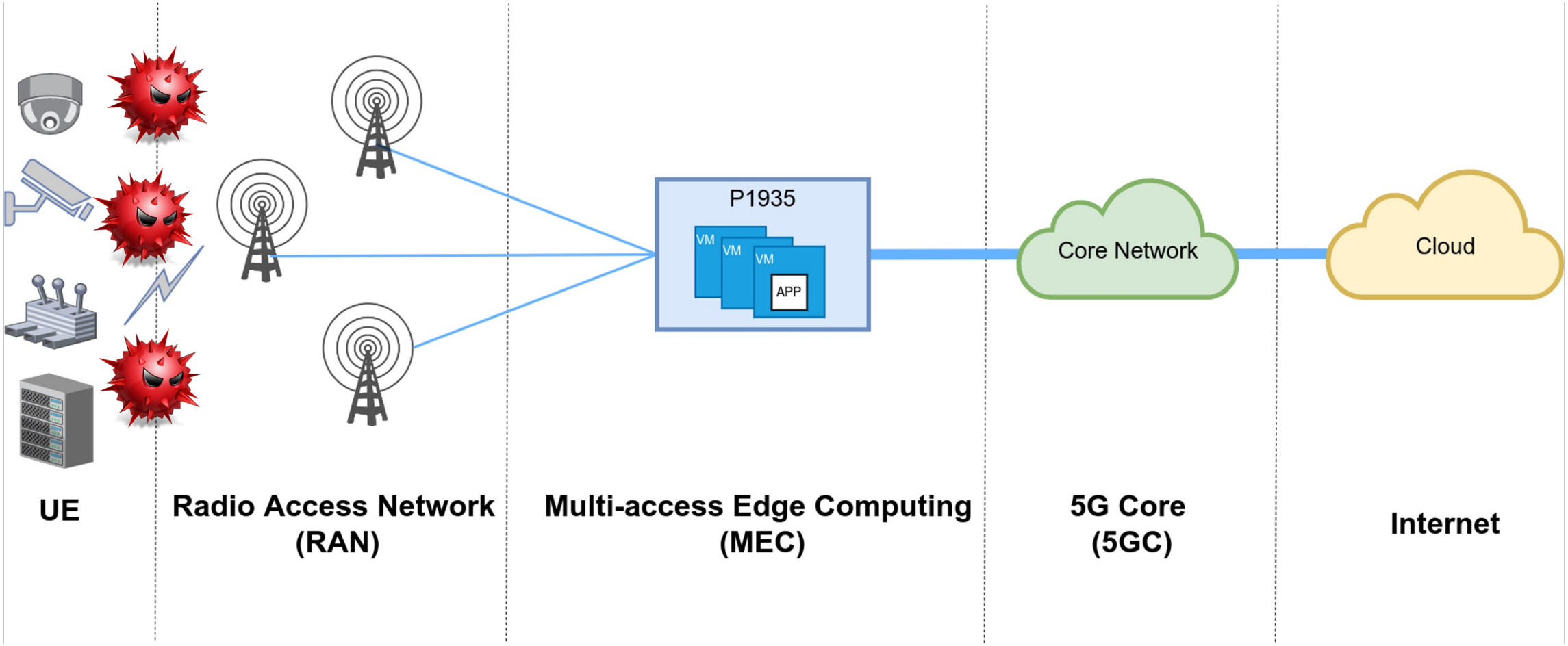
NGAP

NAS

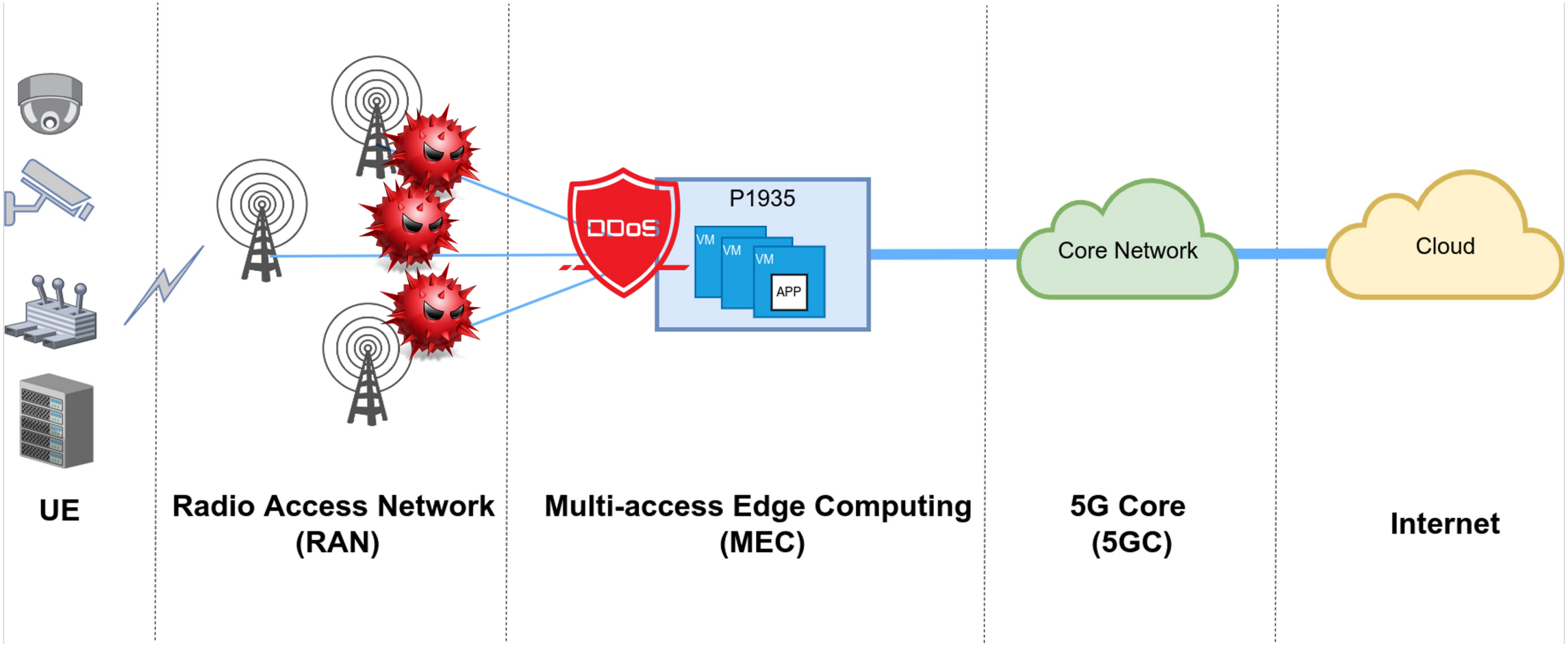
GTP

# DDoS Detection in RAN

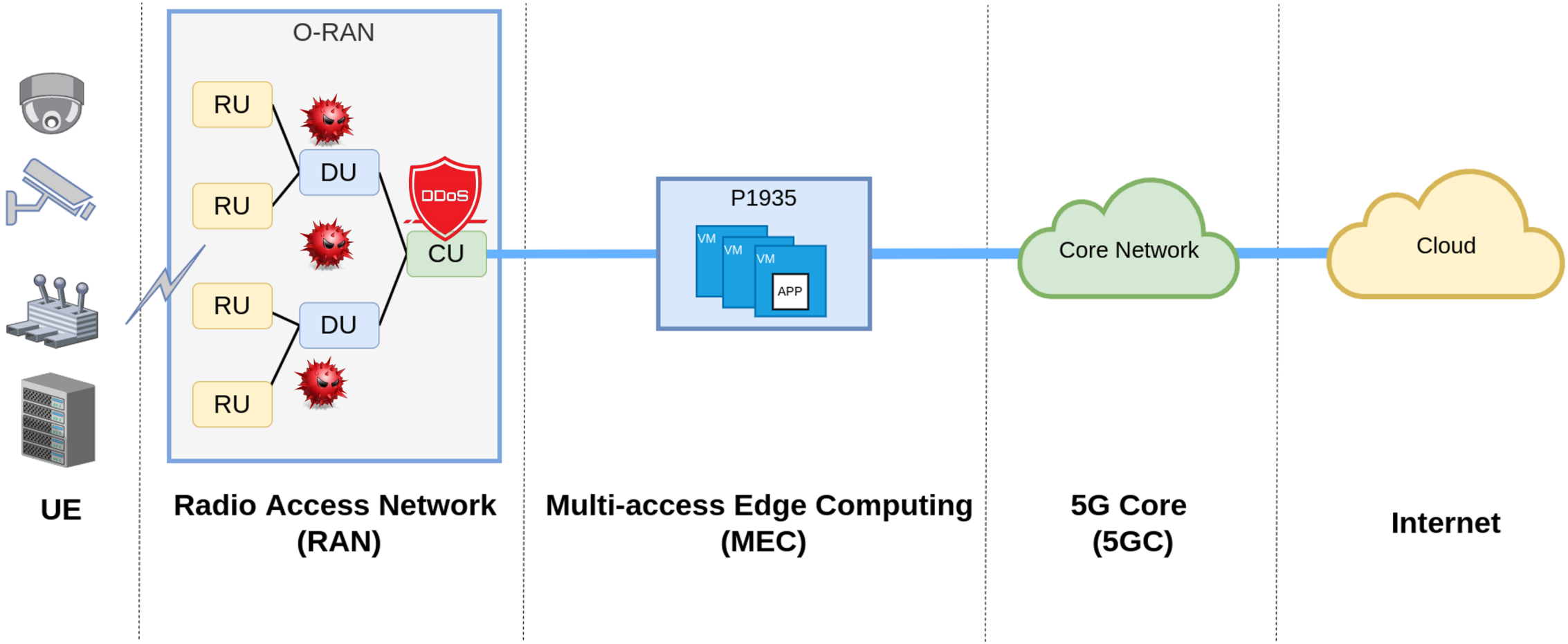
# DDoS Detection at F1 Interface with P1935



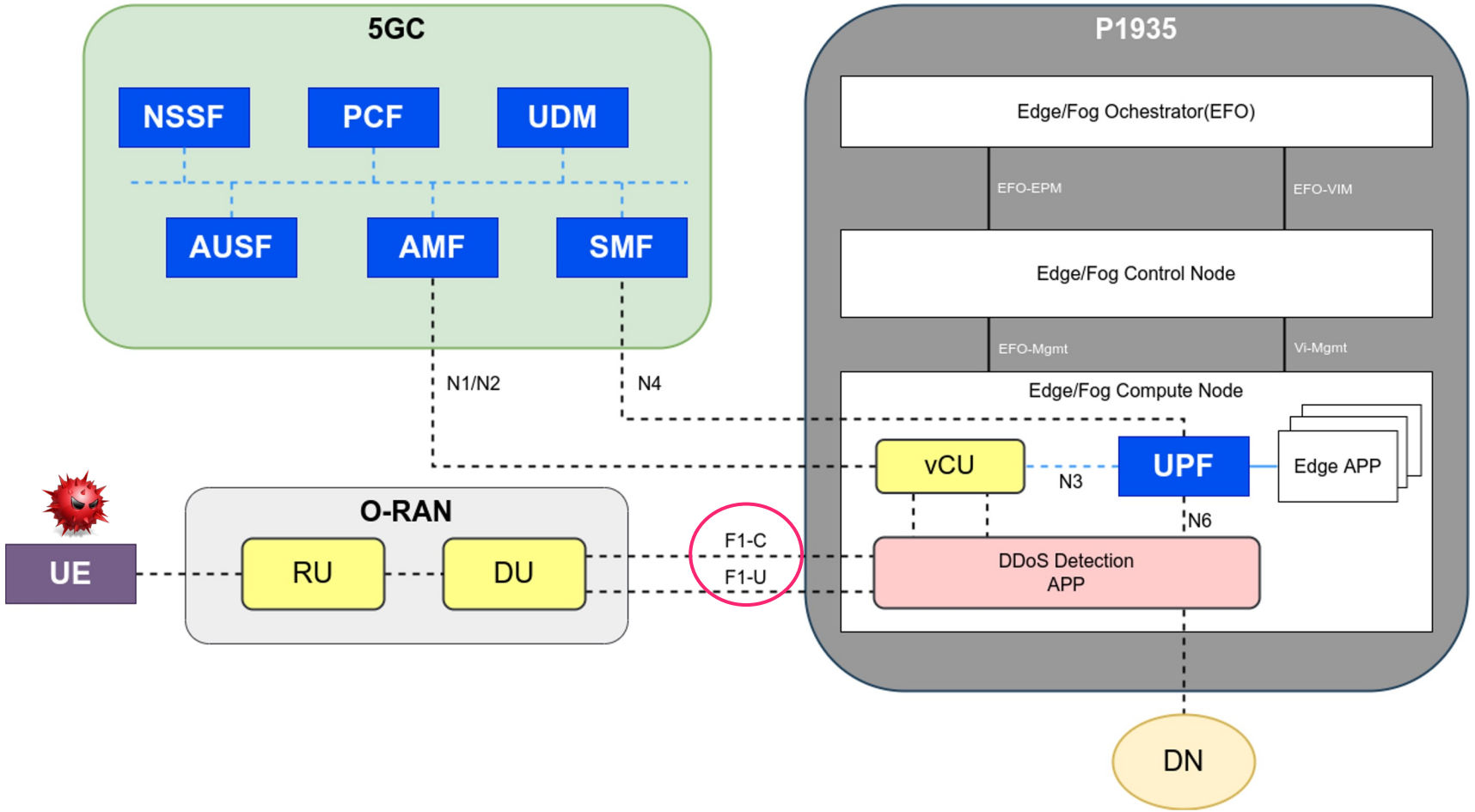
# DDoS Detection at F1 Interface with P1935



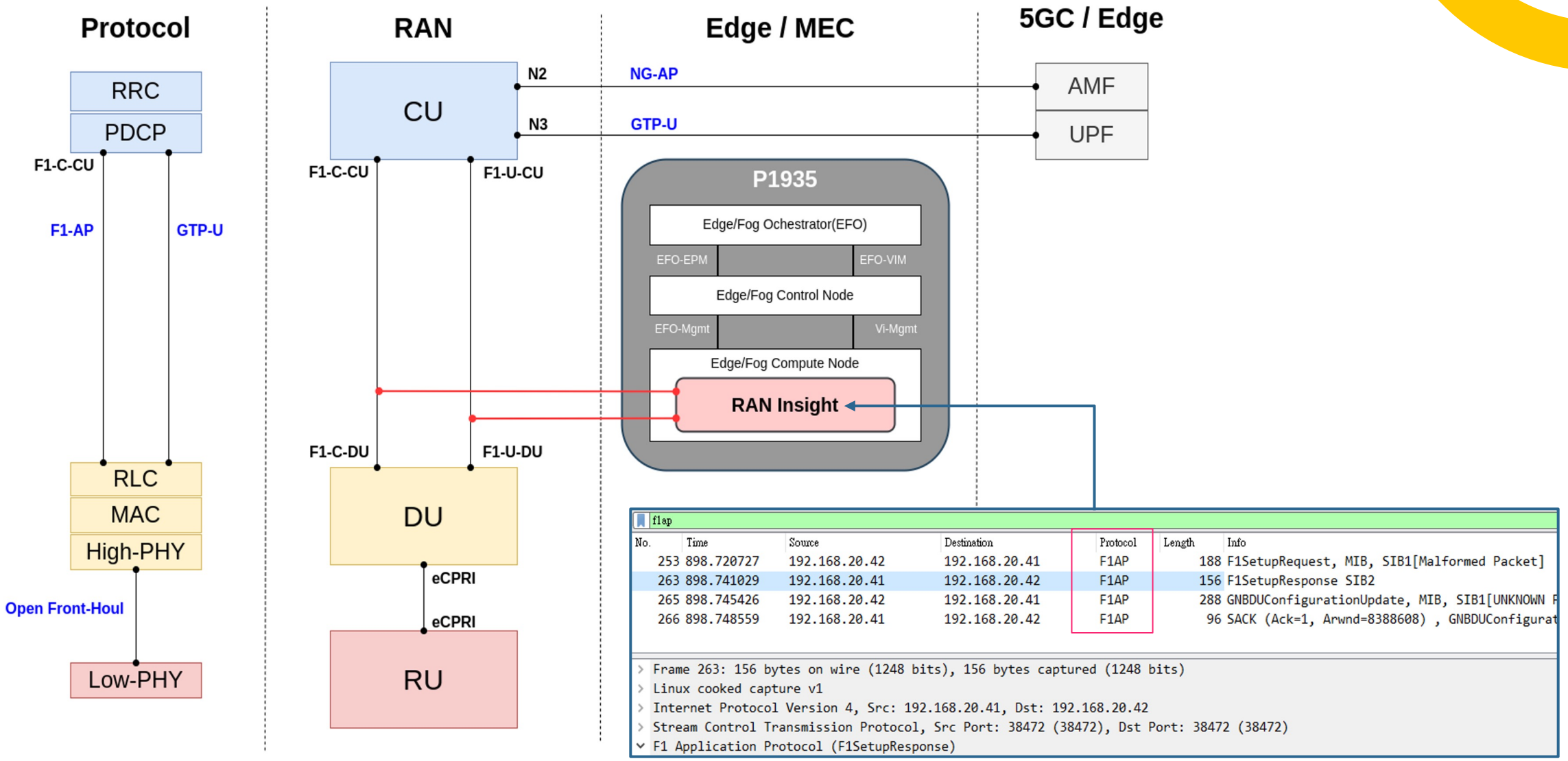
# DDoS Detection at F1 Interface with P1935



# DDoS Detection at F1 Interface with P1935



# Capture Package by F1-C-CU in O-RAN



# F1-AP PDU Decoding

```
▾ ProtocolIE-SingleContainer
    id: id-GNB-DU-Served-Cells-Item (43)
    criticality: reject (0)
  ▾ value
    ▾ GNB-DU-Served-Cells-Item
      ▾ served-Cell-Information
        ▾ nRCGI
          ▾ pLMN-Identity: 64f666
              Mobile Country Code (MCC): Unassigned (466)
              Mobile Network Code (MNC): Unknown (66)
            nRCellIdentity: 0000000010 [bit length 36, 4 LSB pad bits, 0000 0000
        nRPCI: 21
        fiveGS-TAC: 1 (0x000001)
        ▾ servedPLMNs: 1 item
          ▾ Item 0
            ▾ ServedPLMNs-Item
              ▾ pLMN-Identity: 64f666
                  Mobile Country Code (MCC): Unassigned (466)
                  Mobile Network Code (MNC): Unknown (66)
              ▾ iE-Extensions: 1 item
                ▾ Item 0
                  ▾ ProtocolExtensionField
                      id: 131
                      criticality: ignore (1)
                    ▾ extensionValue
                      ▾ SliceSupportList: 1 item
                        ▾ Item 0
                          ▾ SliceSupportItem
                            ▾ sNSSAI
                                sST: 01
                                sD: 010203
        ▾ nR-Mode-Info: tDD (1)
          ▾ tDD
            ▾ nRFreqInfo
                nRARFCN: 720048
              ▾ freqBandListNr: 1 item
```

```
ironman@ubnt22(15:09:42)[~/drive/NTU/Research/nfstream]: python3 payload.py
192.168.44.170 192.168.44.141 64338 16 1754
192.168.20.42 192.168.20.41 546232 122 7564
```

F1SetupRequest

```
F1AP PDU: 450200ac0001400040849027c0a8142ac0a81429964896487d98c19c00000000003008be455a79100
0000000000003e00010077000004004e00020002002a000680000000000001002c00520000002b004c4a0064f66600
0000001000150000010864f66600000083400700004020010203410 00afcb000004e002e0001000000008b400101
000379000412700314a60041466330000008000000000fff000ab000a80000000c7400310020000
```

```
[{'id': 43, 'criticality': 'reject', 'value': ('GNB-DU-Served-Cells-Item', {'served-Cell-Inf
ormation': {'nRCGI': {'pLMN-Identity': b'd\xf6f', 'nRCellIdentity': (1, 36)}, 'nRPCI': 21, '
fiveGS-TAC': b'\x00\x00\x01', 'servedPLMNs': [{'pLMN-Identity': b'd\xf6f', 'iE-Extensions':
[{'id': 131, 'criticality': 'ignore', 'extensionValue': ('SliceSupportList', [{'sNSSAI': {'s
ST': b'\x01', 'sD': b'\x01\x02\x03'}}])}]}], 'nR-Mode-Info': ('tDD', {'nRFreqInfo': {'nRARFC
N': 720048, 'freqBandListNr': [{'freqBandIndicatorNr': 79, 'supportedSULBandList': []}]}, 't
ransmission-Bandwidth': {'nRSCS': 'scs30', 'nRNRB': 'nrb273'}}), 'measurementTimingConfigura
tion': b'\x00', 'iE-Extensions': [{'id': 139, 'criticality': 'ignore', 'extensionValue': ('R
ANAC', 1)}]}, 'gNB-DU-System-Information': {'mIB-message': b'y\x00\x04', 'sIB1-message': b'p
\x03\x14\xa6\x00AFc0\x00\x00\x08\x00\x00\x00\x00\xff\xf0'}})}]
```

F1SetupResponse

```
F1AP PDU: 4502008c37d1400040845877c0a81429c0a8142a9648964885bfa58e0000000000300697eb40b7800
0000000000003e40010055000003004e000200020003003a0000000440342064f66600000000100000007600250 0
00213c3850a15424a203c8929bdffe15b685011a1046851808f804d243ecc040c036000000aa000a80000000c700
03100200000000
```

```
[{'id': 4, 'criticality': 'ignore', 'value': ('Cells-to-be-Activated-List-Item', {'nRCGI': {
'pLMN-Identity': b'd\xf6f', 'nRCellIdentity': (1, 36)}, 'iE-Extensions': [{'id': 118, 'criti
cality': 'reject', 'extensionValue': ('GNB-CUSystemInformation', {'sibtypetobeupdatedlist':
[{'sIBtype': 2, 'sIBmessage': b'<8P\xa1T$\xa2\x03\xc8\x92\x9b\xdf\xfe\x15\xb6\x85\x01\x1a\x1
0F\x85\x18\x08\xf8\x04\xd2C\xec\xc0@\xc06\x00', 'valueTag': 0}]})}]})}]
{'latest-RRC-Version': (0, 3), 'iE-Extensions': [{'id': 199, 'criticality': 'reject', 'exten
sionValue': ('OCTET STRING', b'\x10\x02\x00')}]}
192.168.20.42 192.168.20.41 62296 25 2676
```

# Conclusion and Future Work

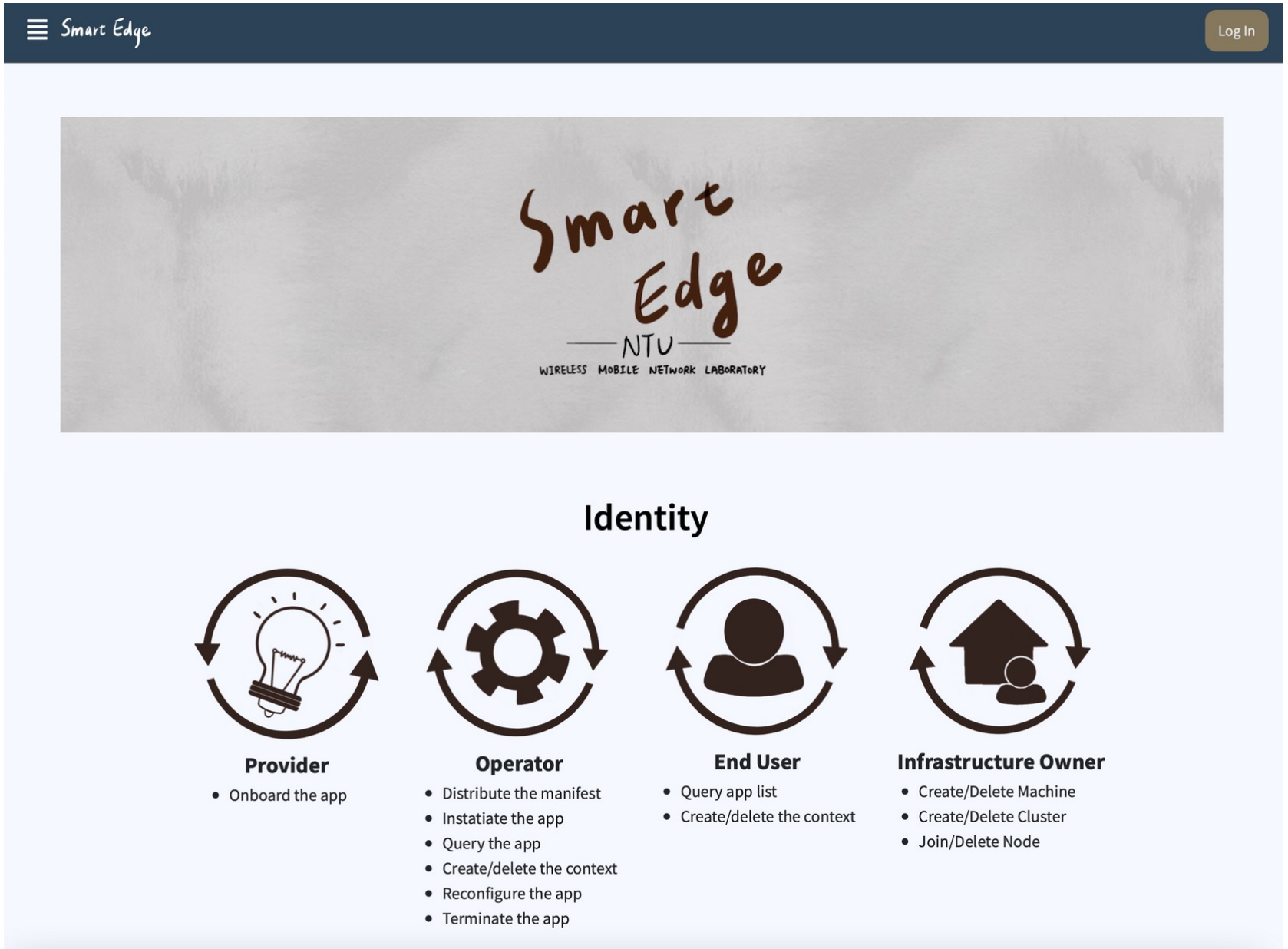
- We built a 5G RAN and Core testbed by UERANSIM and Open5GS.
- We designed an F1-AP protocol analytic tool by using python's libraries: "NFStream," "scapy," and "pycrate," then parsed the F1AP pcapng file to the dataset succeeded.
- Need to find RRC DDoS tools or make an RRC simulator tool by self.

# UI & Website

Presenter: 李婕妤

WIRELESS MOBILE NETWORK LABORATORY

# User Interface



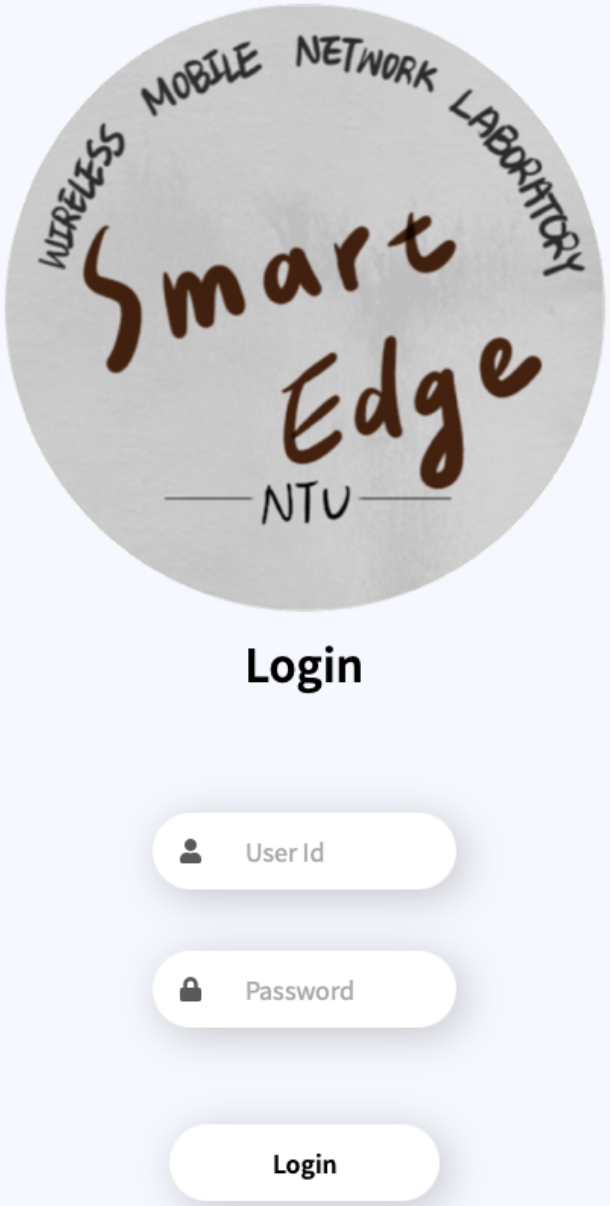

# Home Page

- Four accounts
- You can see what each account can do on the homepage.

wireless mobile network laboratory

# Login

Each user uses a different account and password to log in.



# Actions

- Resource management
  - Physical machine
  - Virtual machine

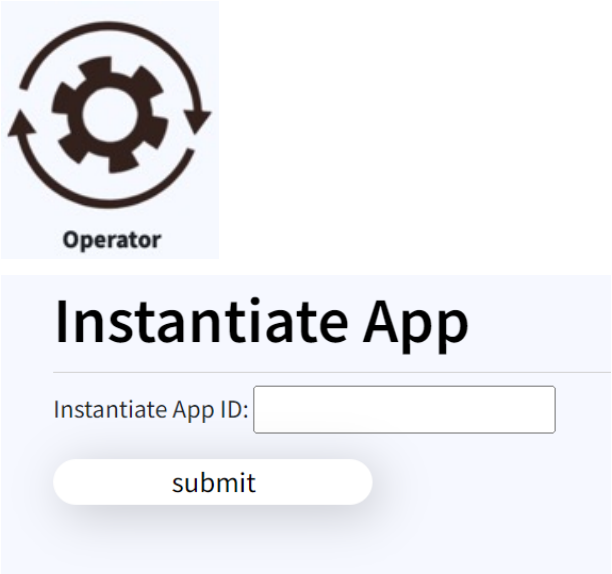
- Application management
  - Run in container (Kubernetes)
  - Configured by manifest

# Resource Management

- Creation
- Query/Discovery
- Reconfiguration
- Deletion

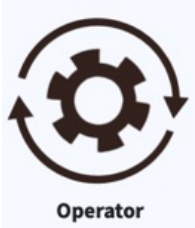
# Application Management (1/2)

- Onboarding
- Instantiation

# Application Management (2/2)

- Context Creation/Deletion
- Reconfiguration
- Termination





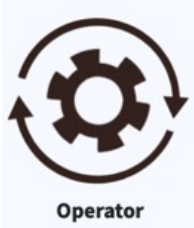
## Create/Delete Context

App ID:

Action:

Command:

submit




## Reconfigure App

Reconfigure App ID:

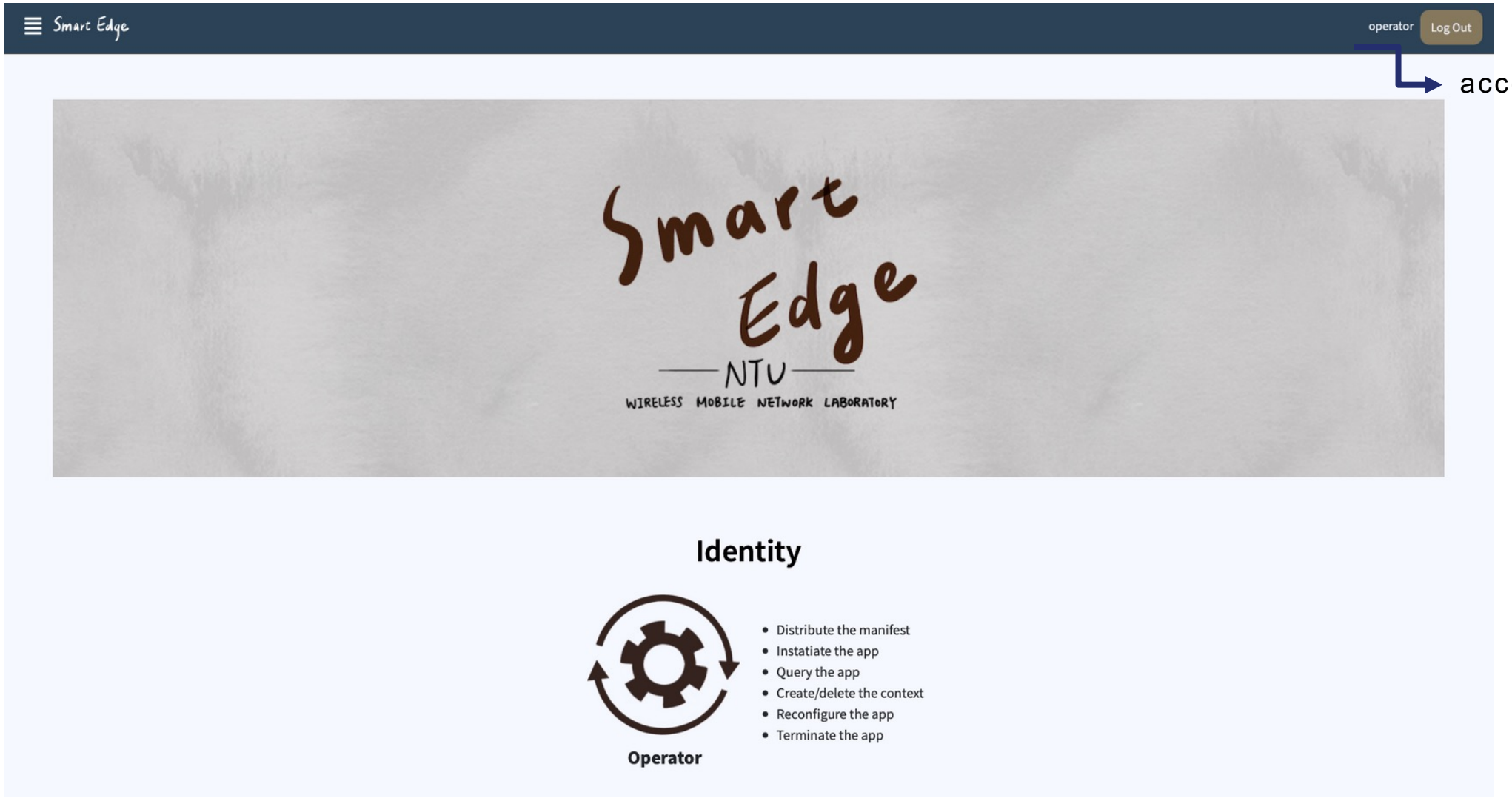
New manifest file: 選擇檔案 未選擇任何檔案 Upload

## Terminate App

Terminate App ID:

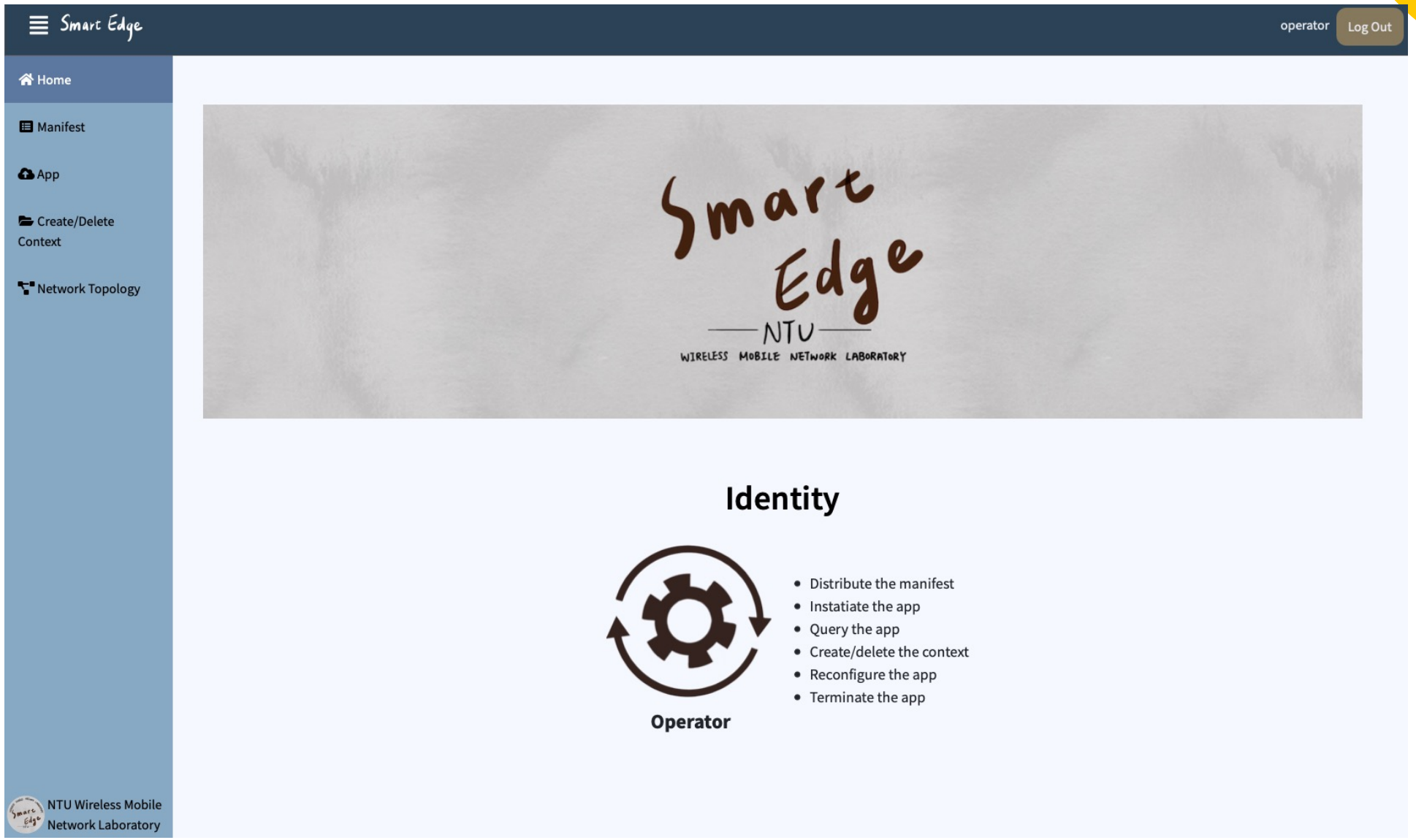
submit

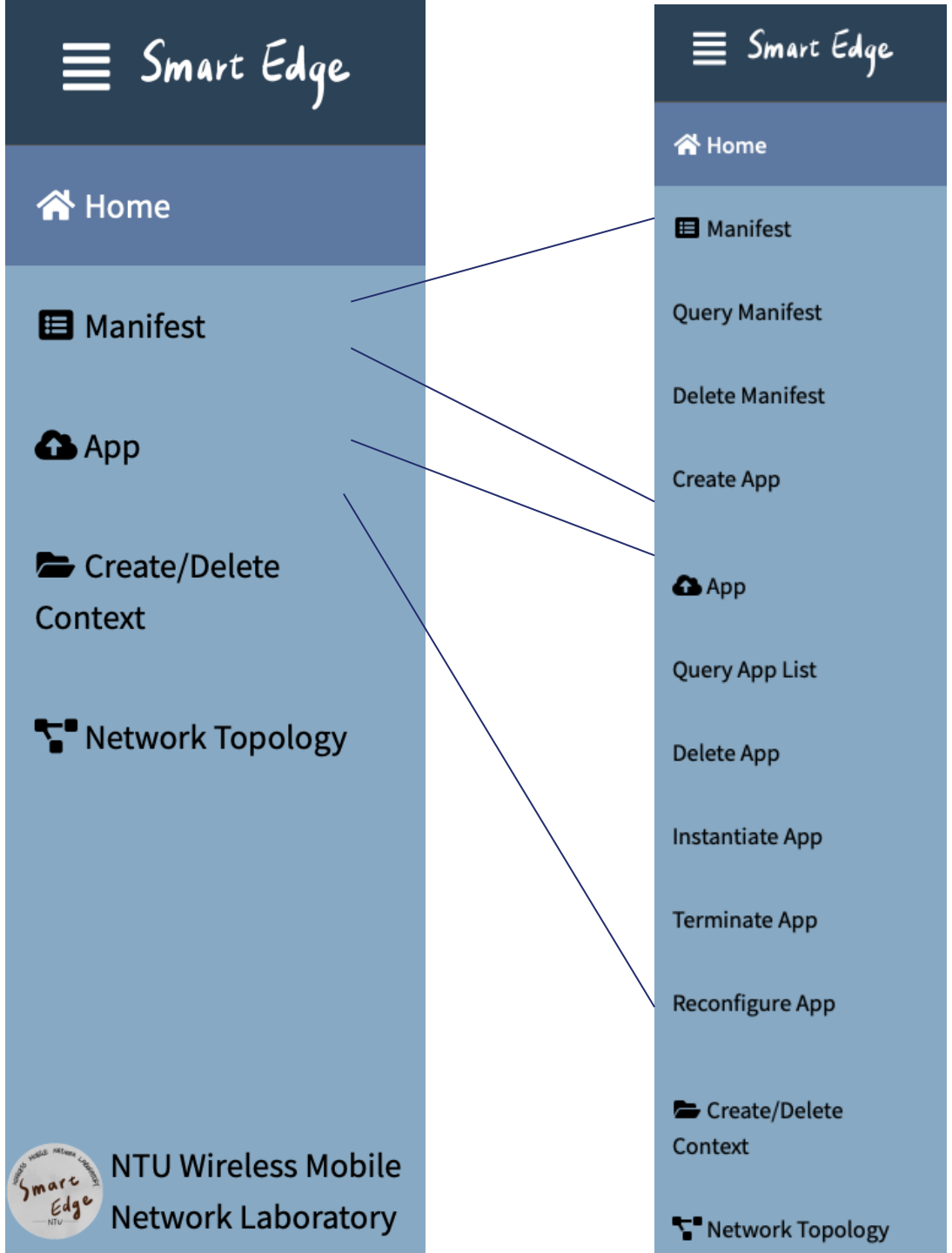
# Home Page (Operator)



# Home Page

Hidden sidebar





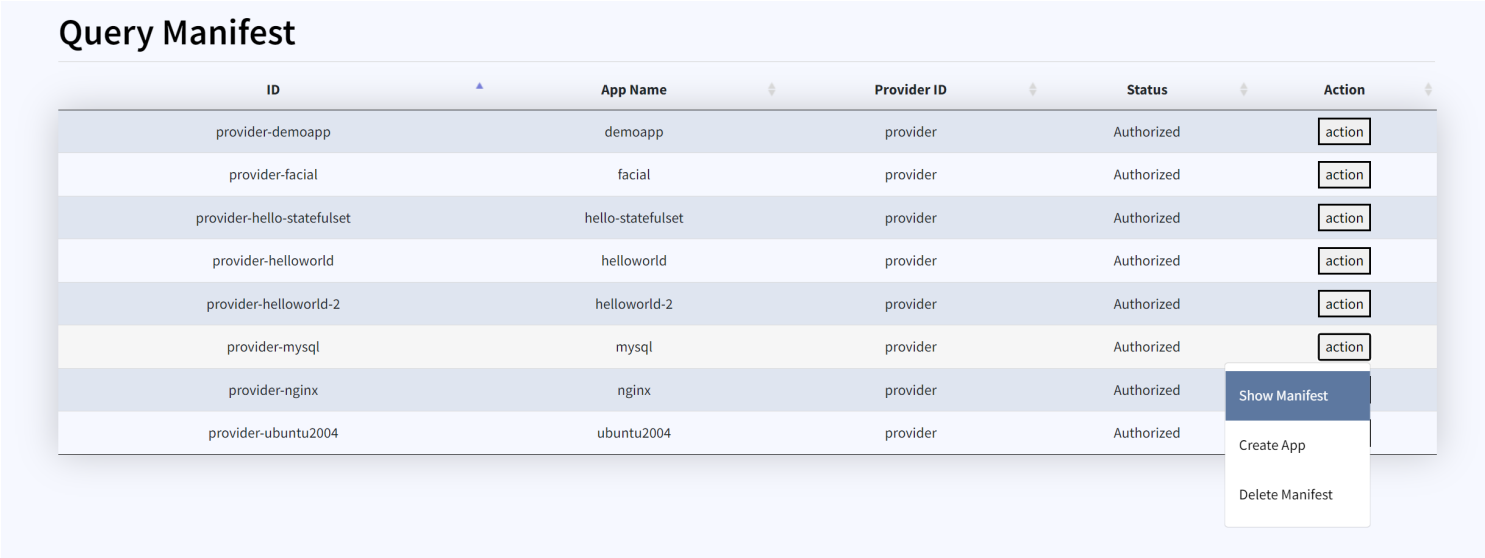

wireless mobile network laboratory

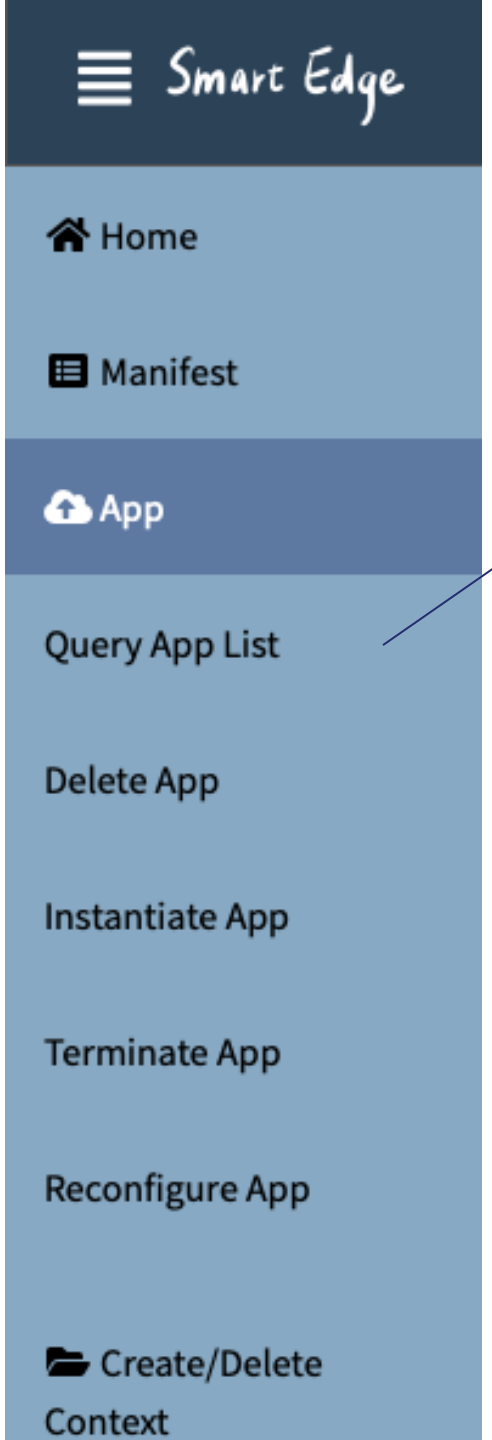
# Sidebar

- All the function clicks are here.
- The user can click on to operate the desired function.


# Manifest

- In this category, users can create app, delete manifest, and view apps uploaded by the service provider.



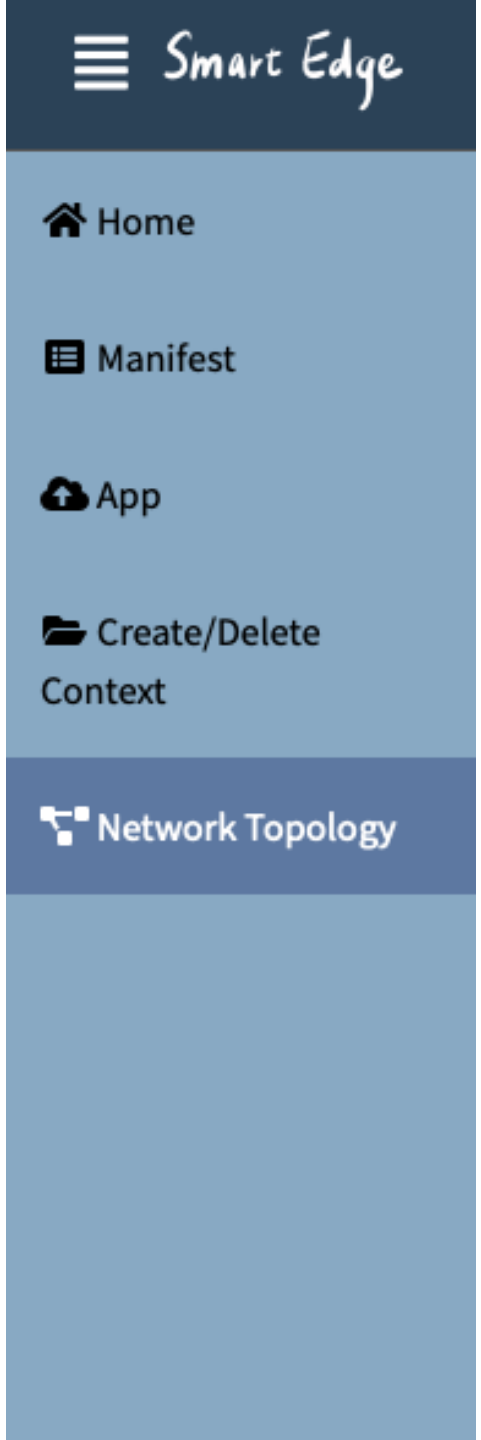



# App

- In this category, users can terminate app, delete app, instantiate app and reconfigure app.

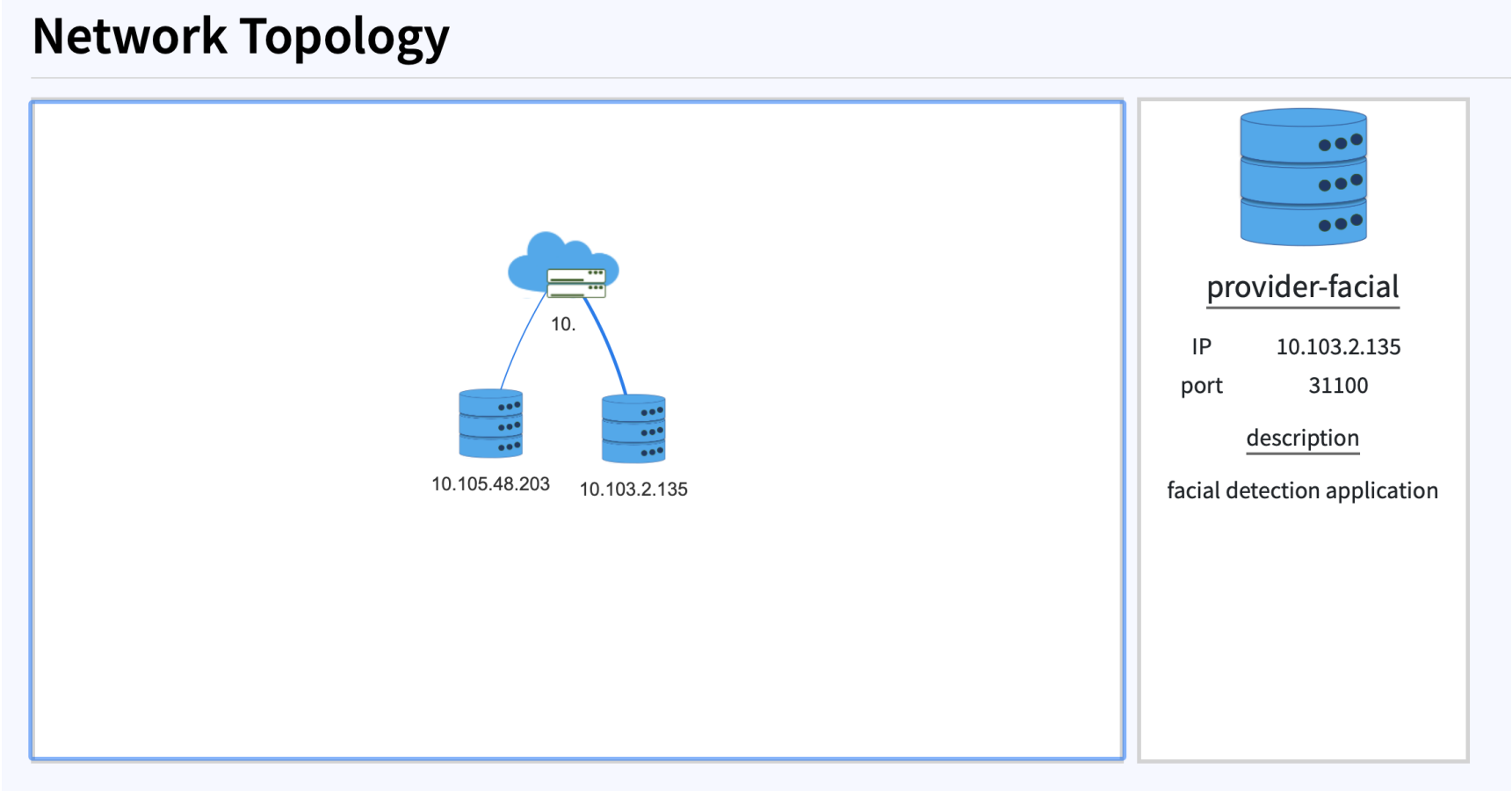
## Query App List

| ID | Auth By | Ref. Manifest ID | Status | Action |
|---|---|---|---|---|
| provider-anno-test | operator | provider-anno-test | 0 | action |
| provider-helloworld | operator | provider-helloworld | 1 | action |
| provider-helloworld-ingress | operator | provider-helloworld-ingress | 0 | action |
| provider-stream | operator | provider-stream | 0 | action |

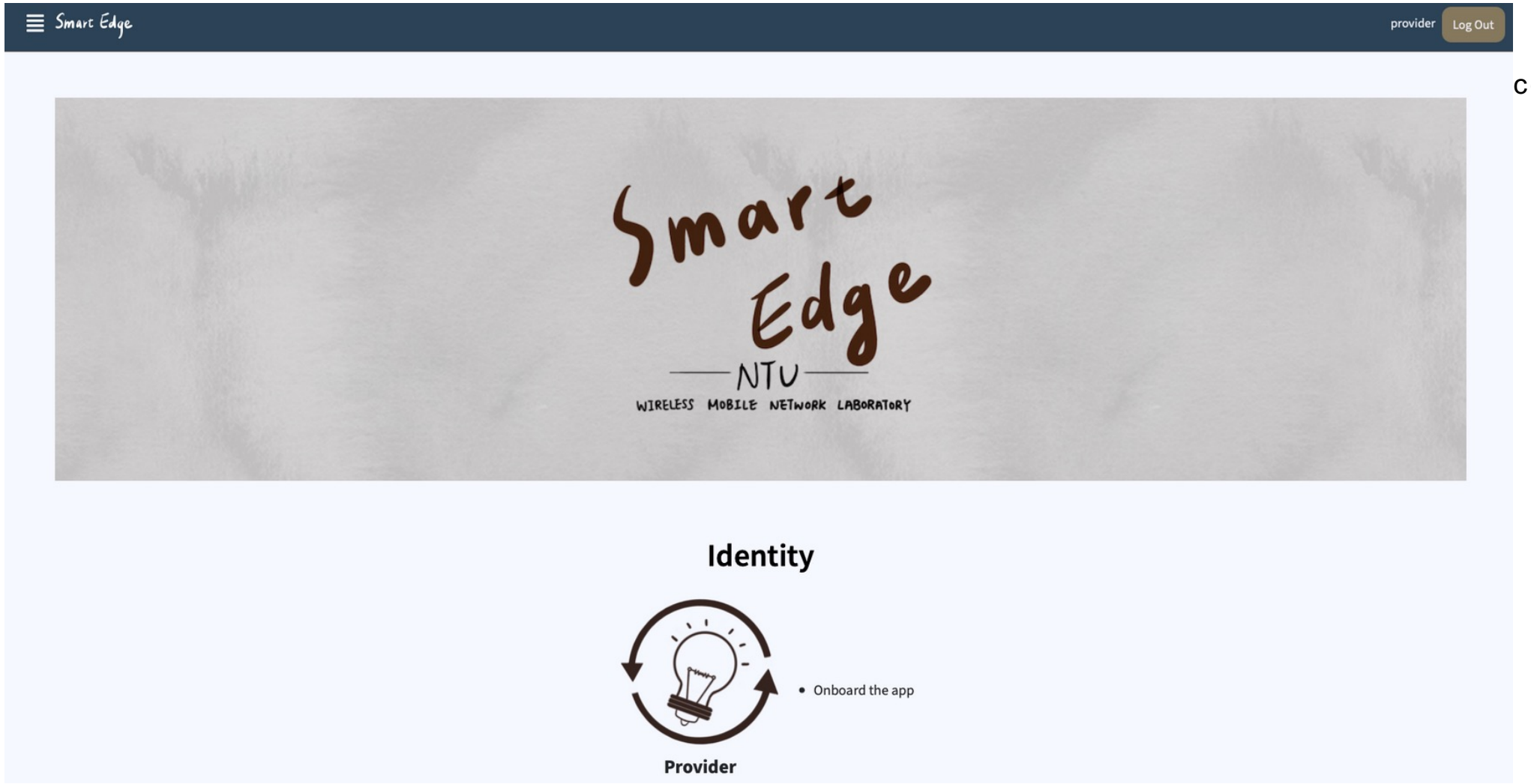



# Network Topology

- service topology



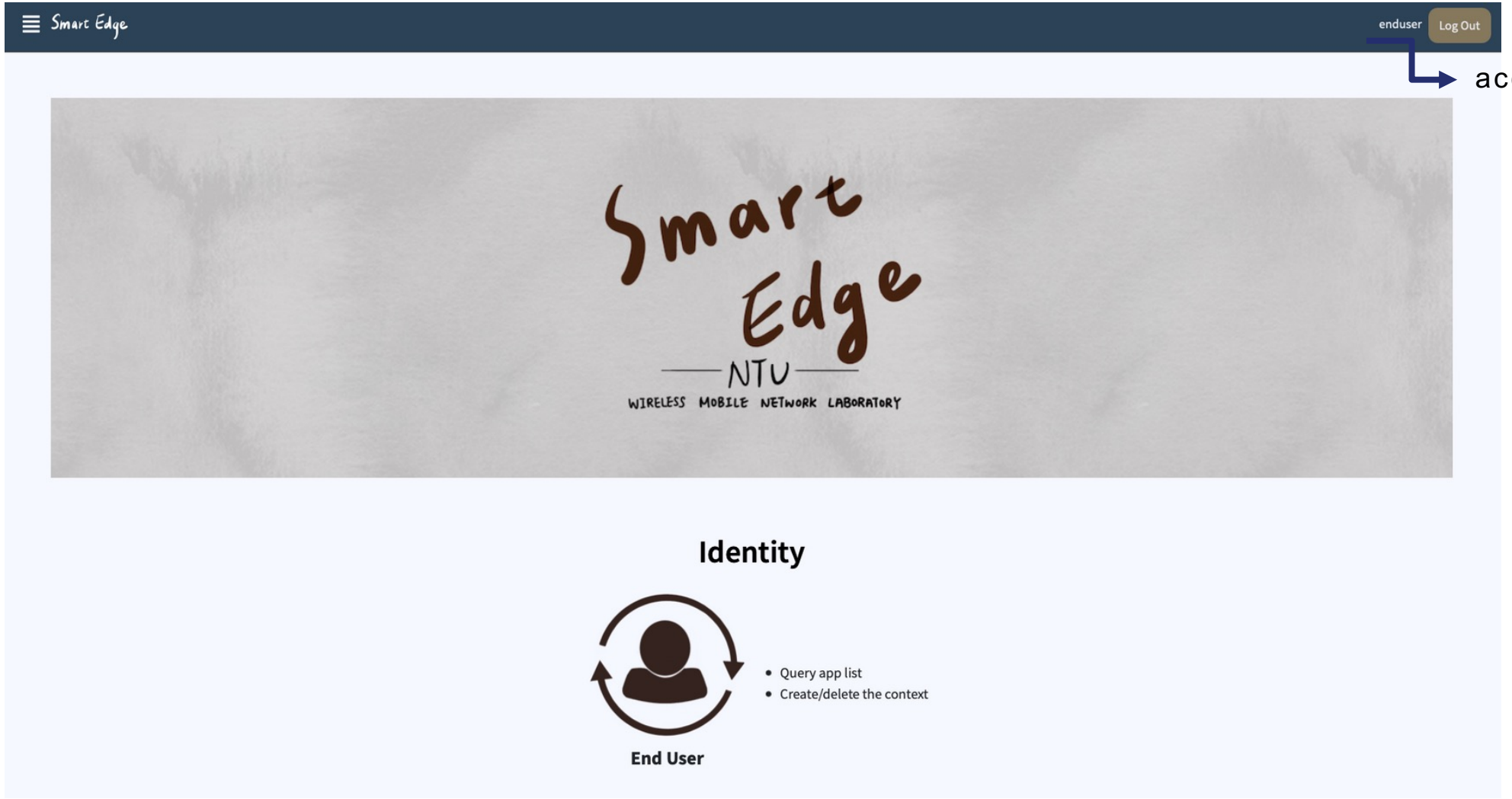
# Home Page (Provider)




count name

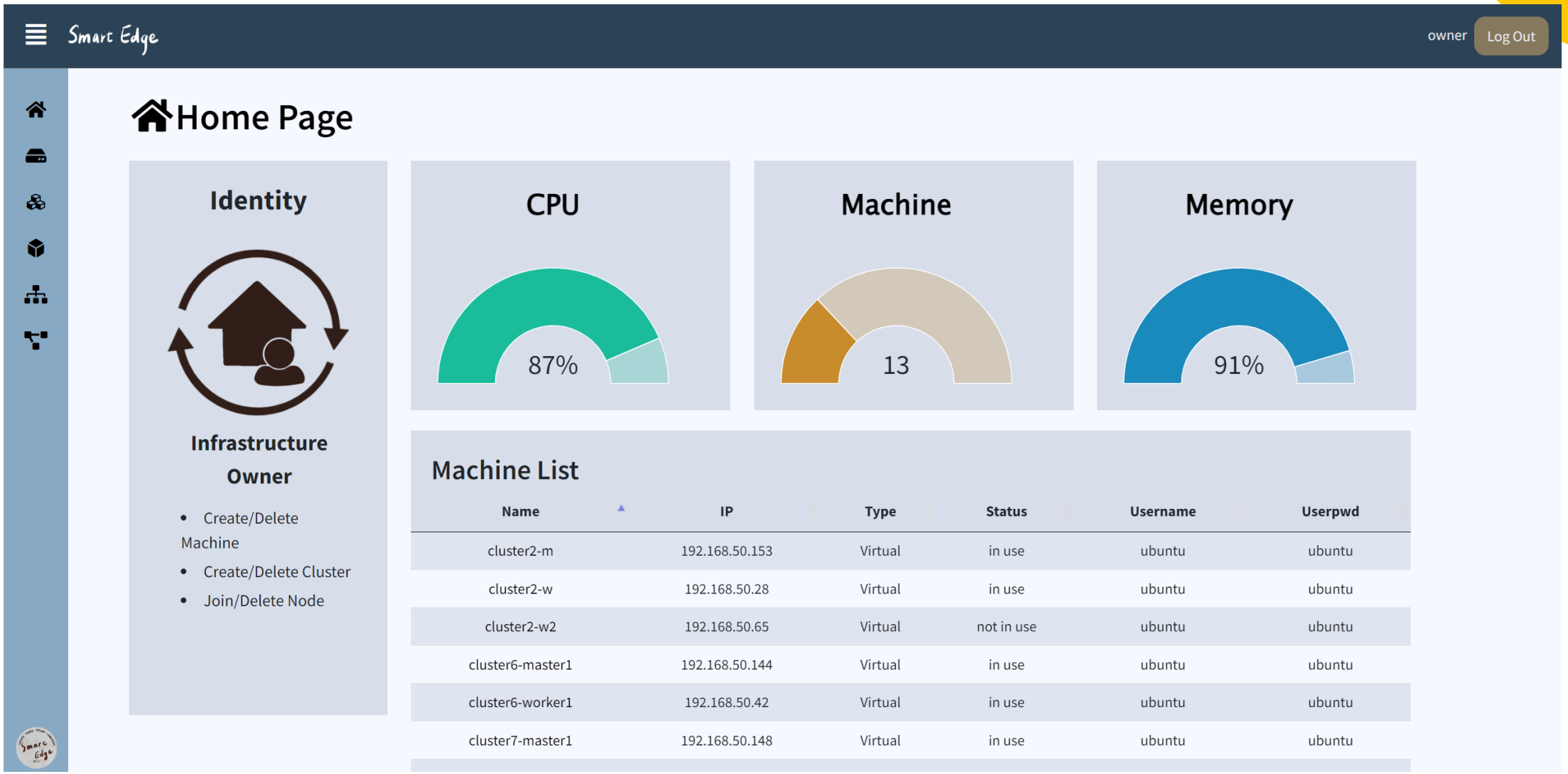
wireless mobile network laboratory

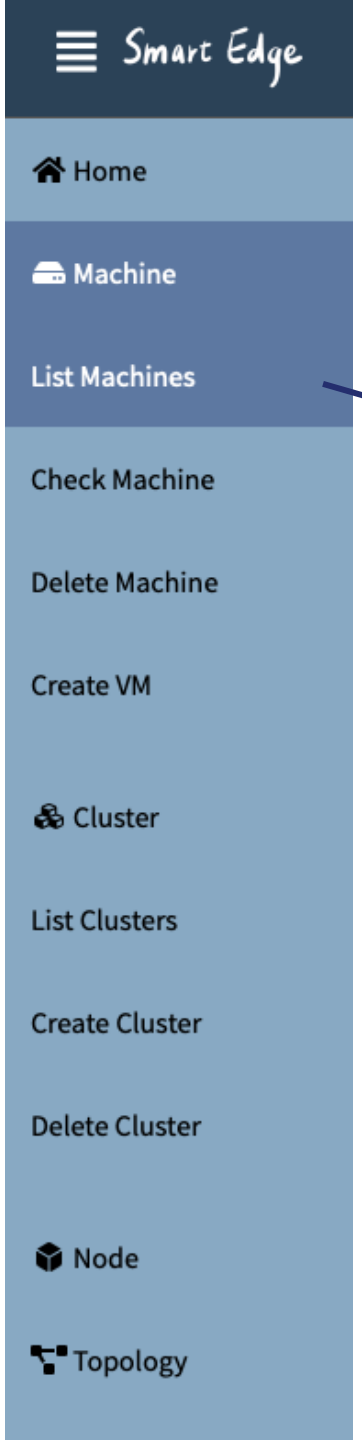
# Home Page (End User)




account name

# Home Page (Owner)






# Machine

- In this category, users can check machine, delete machine, create VM and view the list of machines.

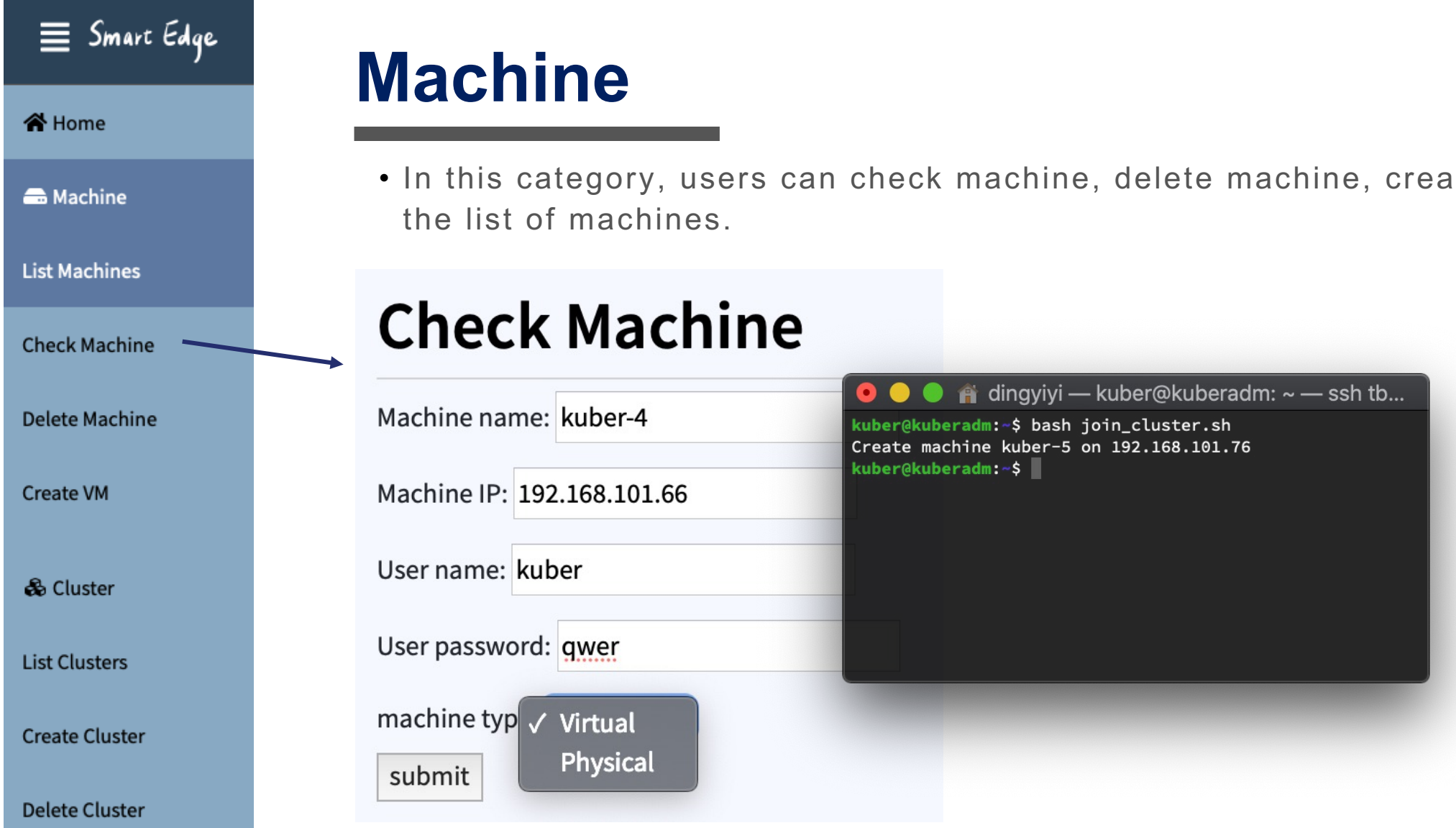
## List Machines

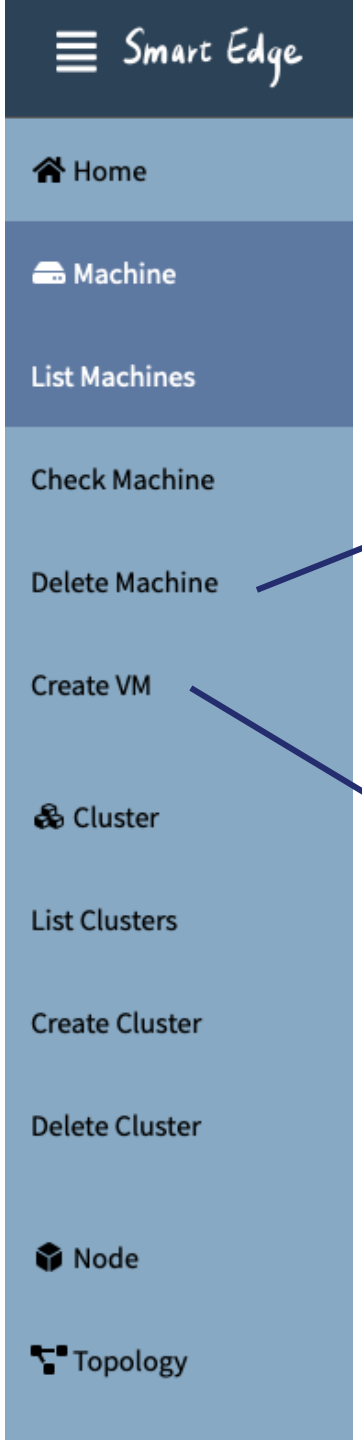
Check Machine | Create VM | Delete Machine

| Name | IP | Status | Type | Username | Userpwd |
|---|---|---|---|---|---|
| cluster2-m | 192.168.50.153 | in use | Virtual | ubuntu | ubuntu |
| cluster2-w | 192.168.50.28 | in use | Virtual | ubuntu | ubuntu |
| cluster6-master1 | 192.168.50.144 | in use | Virtual | ubuntu | ubuntu |
| cluster6-worker1 | 192.168.50.42 | in use | Virtual | ubuntu | ubuntu |
| cluster7-master1 | 192.168.50.148 | in use | Virtual | ubuntu | ubuntu |
| cluster7-worker1 | 192.168.50.31 | in use | Virtual | ubuntu | ubuntu |
| cluster7-worker2 | 192.168.50.194 | in use | Virtual | ubuntu | ubuntu |
| cluster_1_master | 192.168.50.242 | not in use | Virtual | ubuntu | ubuntu |
| cluster_1_worker | 192.168.50.236 | not in use | Virtual | ubuntu | ubuntu |
| EPC VM | 192.168.50.141 | in use | EPC | | |
| Fii BS | 192.168.50.70 | in use | BS | | |

# Machine

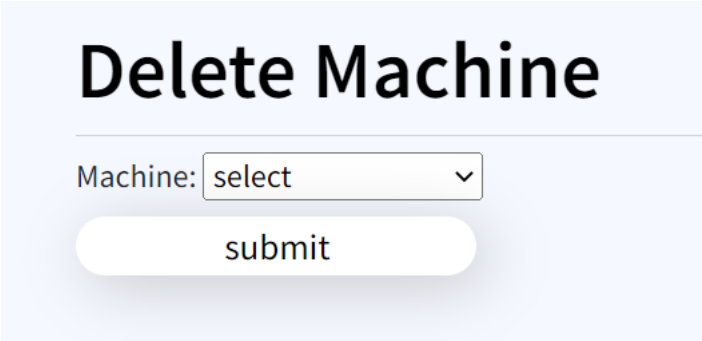
- In this category, users can check machine, delete machine, create VM and view the list of machines.

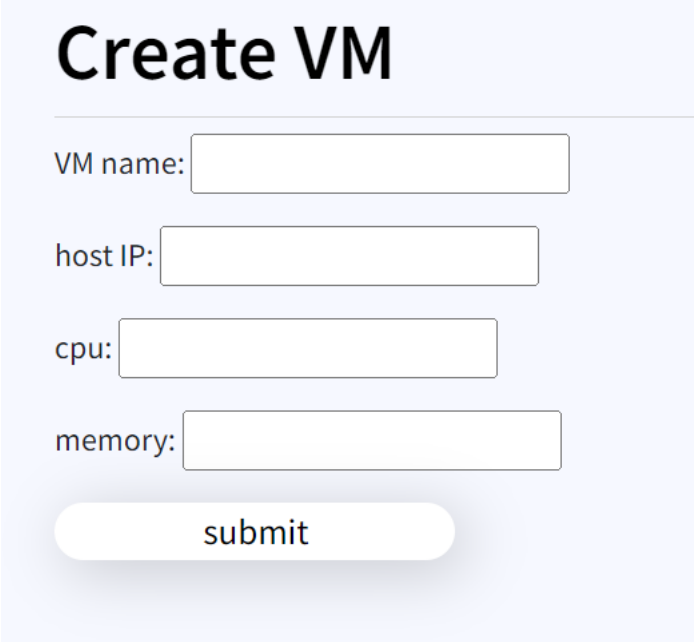


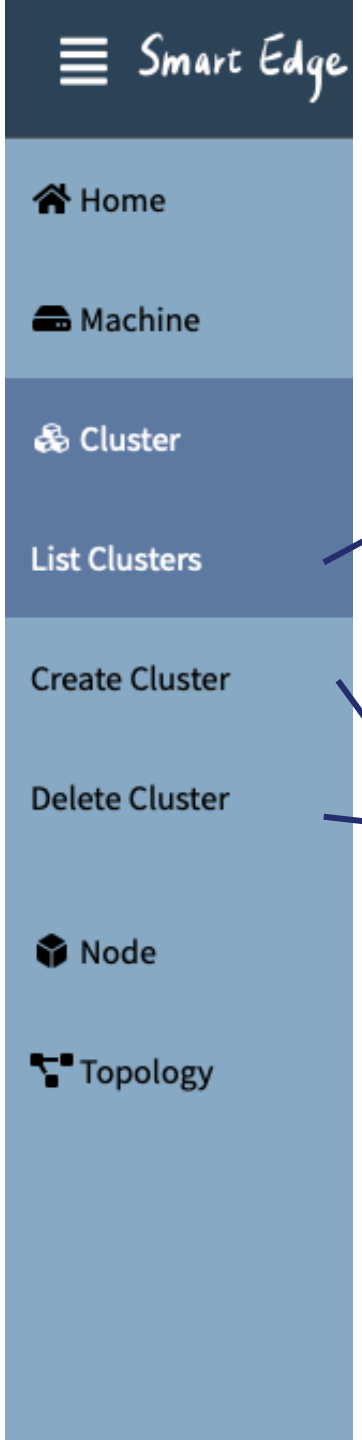



# Machine

- In this category, users can check machine, delete machine, create VM and view the list of machines.






# Cluster

- In this category, users can create cluster, delete cluster and view the list of clusters.

## List Clusters

Create Cluster | Delete Cluster

| Name | Master Node | Join CMD |
|---|---|---|
| cluster2 | cluster2-m | kubeadm join 192.168.50.153:6443 --token 8l2ong.kde19ox2hfi7w07b --discovery-token-ca-cert-hash sha256:0a5efcd46a6476ec5e7bf939271491f889a6356d86e26a5472b3cbe182a90688 |
| cluster6 | cluster6-master1 | kubeadm join 192.168.50.144:6443 --token 4n80lp.0pddk1ekomt3zajg --discovery-token-ca-cert-hash sha256:1449879adf122bb587cab345a4ba0e15e87ce693c869d7d9d44d2eaf8fff9d0a |
| cluster7 | cluster7-master1 | kubeadm join 192.168.50.148:6443 --token h7j2ev.2kmjvlle4g44d6cx --discovery-token-ca-cert-hash sha256:3c509962e7c214daafc40c5badf5f54af16c9335207bc176c21bef30d58a78ad |

## Create Cluster

Cluster name:

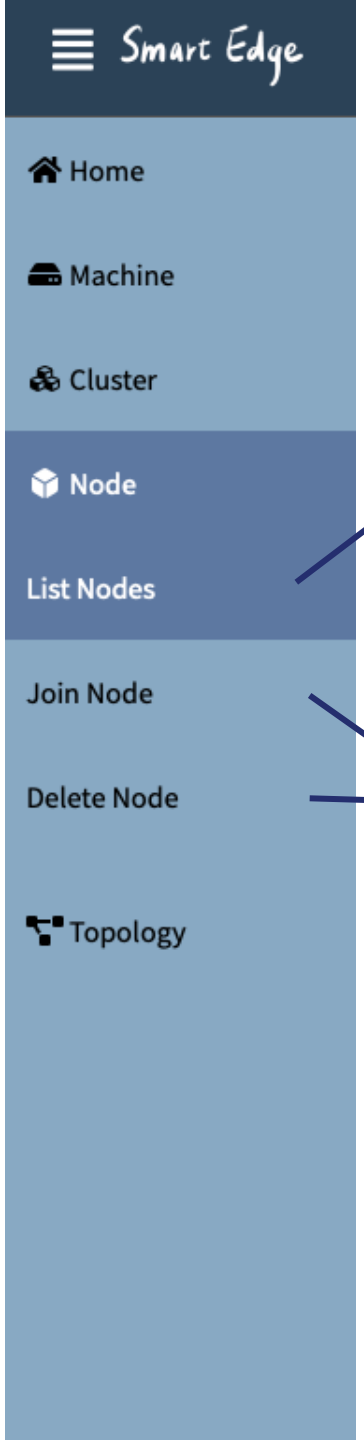
Machine: select

submit

## Delete Cluster

Cluster name: select

submit




# Node

- In this category, users can join node, delete node and view the list of nodes.

## List Nodes

cluster_1 | Join Node | Delete Node

| Name | Address | Available CPU | Creation Time | Master |
|---|---|---|---|---|
| ubuntu1804 | 192.168.50.242 | 2 | Thu, 01 Jul 2021 07:15:37 GMT | true |
| 192.168.50.236 | 192.168.50.236 | 2 | Thu, 01 Jul 2021 07:32:13 GMT | false |

## Join Node

Cluster: select

Machine: select

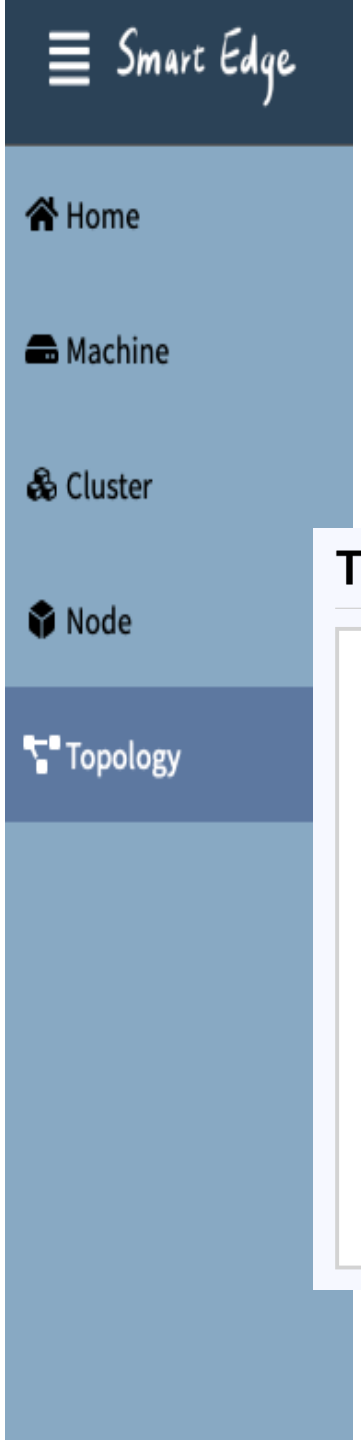
submit

## Delete Node

Cluster: select

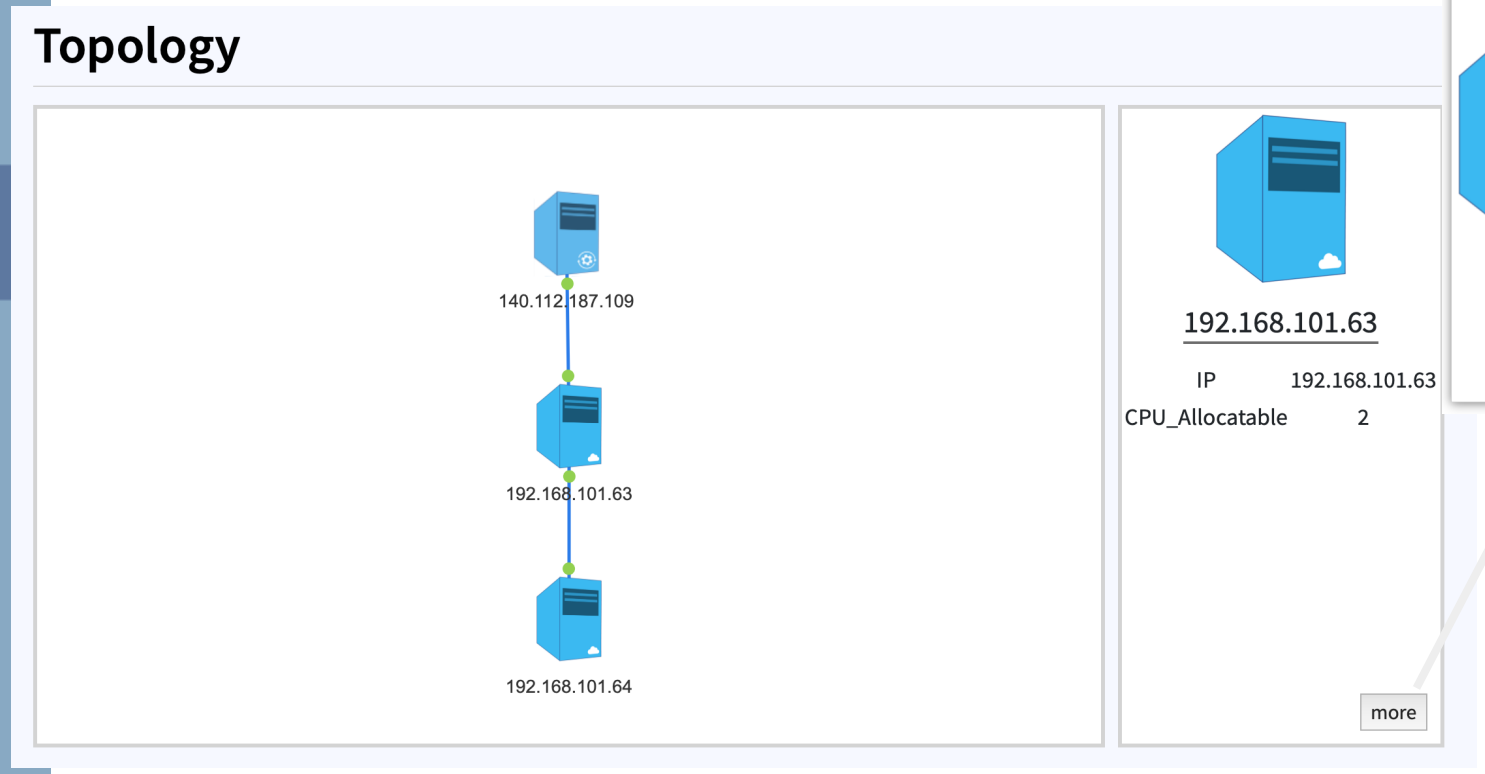
Node name:

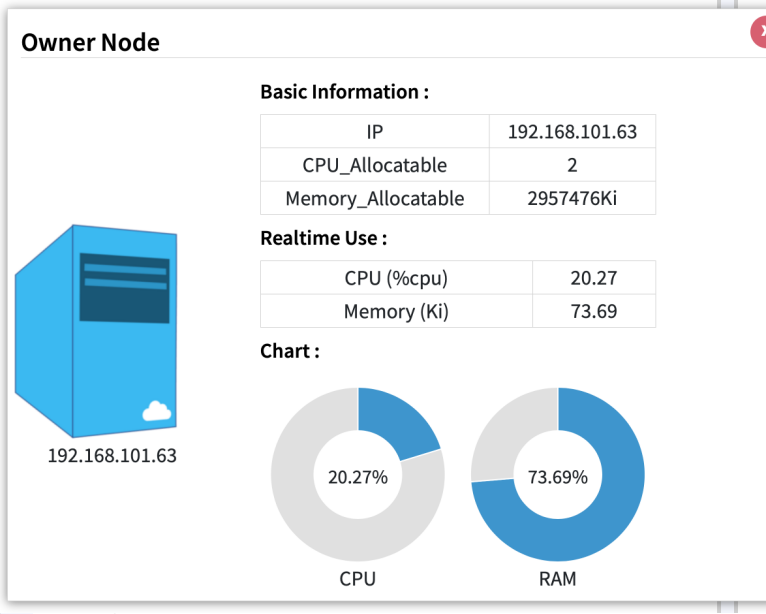
submit




# Network Topology

- service topology



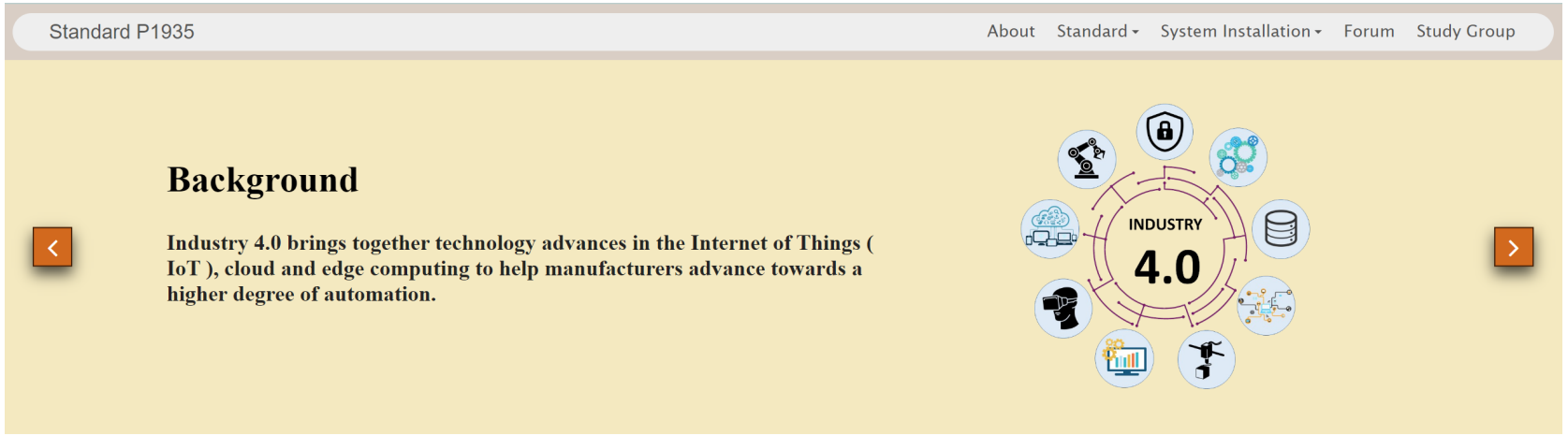

Detailed information

WIRELESS MOBILE NETWORK LABORATORY

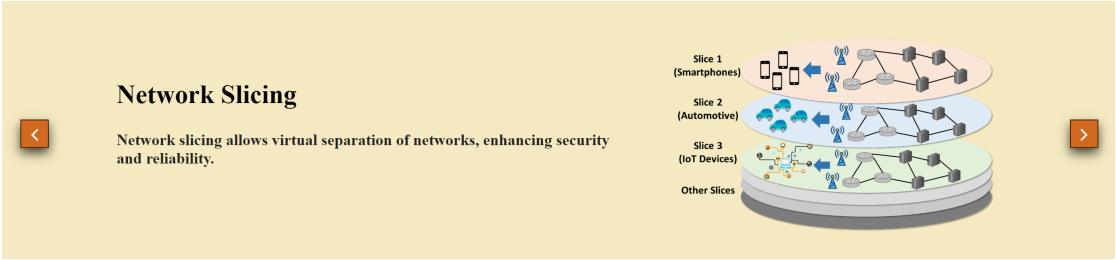
# Standard P1935 Website

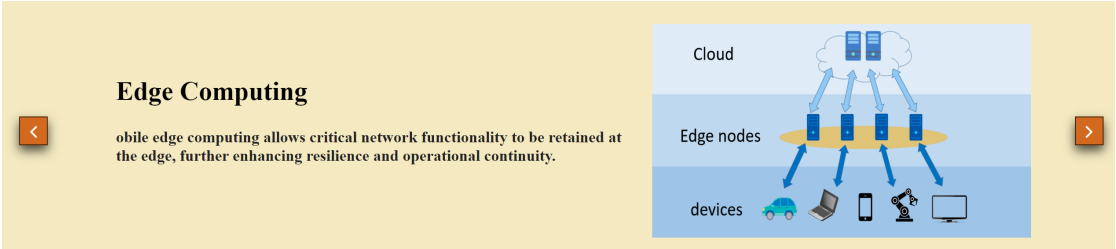
# Home Page

- Background information

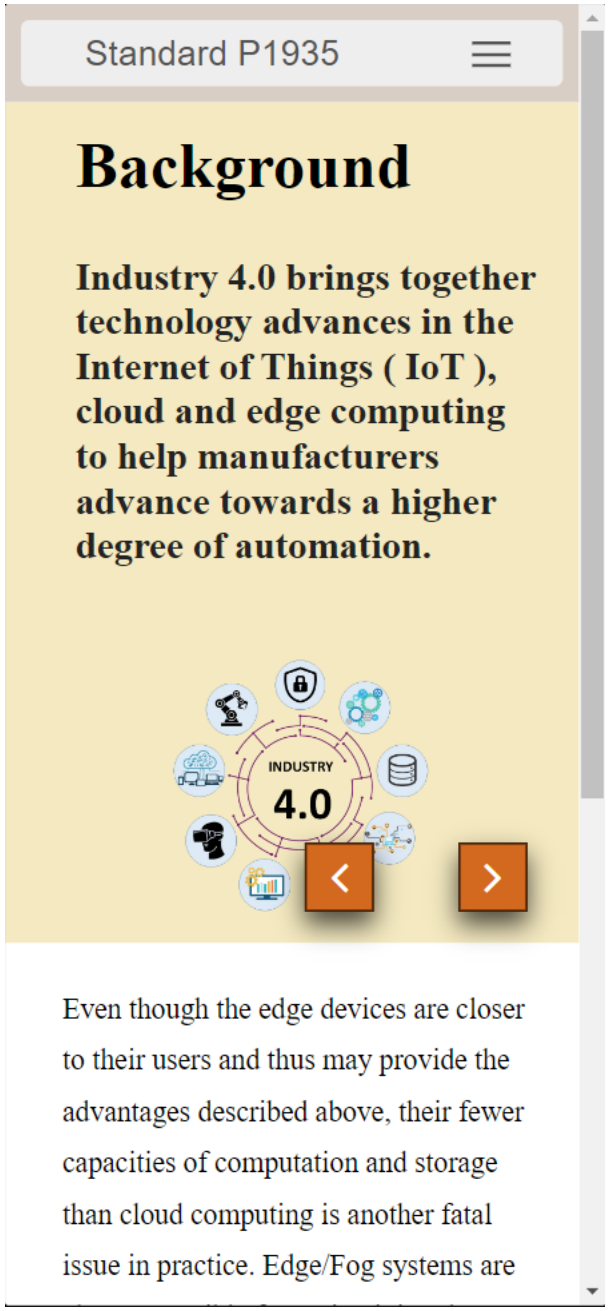


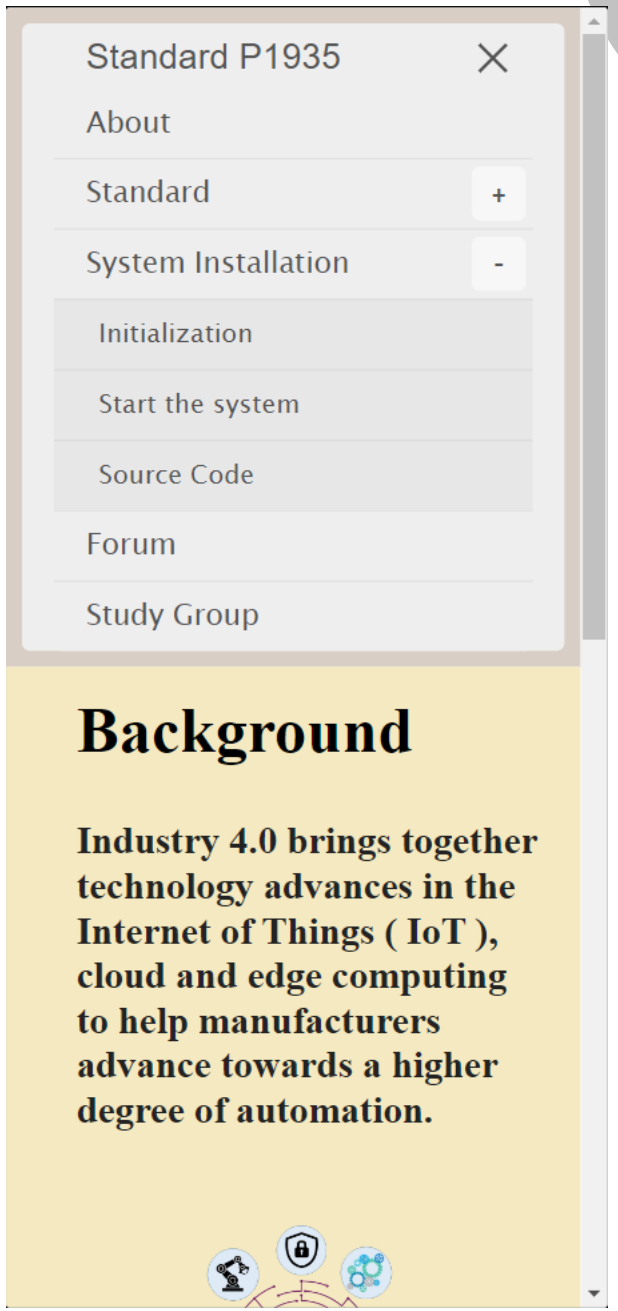

Even though the edge devices are closer to their users and thus may provide the advantages described above, their fewer capacities of computation and storage than cloud computing is another fatal issue in practice. Edge/Fog systems are also responsible for maintaining the common operations, handling the application lifecycle, providing a corresponding environment for the mobile services, performing the functionalities of storage and traffic control, and supporting service mobility.



# Home Page


Standard P1935
Background
Industry 4.0 brings together technology advances in the Internet of Things ( IoT ), cloud and edge computing to help manufacturers advance towards a higher degree of automation.
INDUSTRY
4.0
Even though the edge devices are closer to their users and thus may provide the advantages described above, their fewer capacities of computation and storage than cloud computing is another fatal issue in practice. Edge/Fog systems are


Standard P1935
About
Standard
+
System Installation
-
Initialization
Start the system
Source Code
Forum
Study Group
Background
Industry 4.0 brings together technology advances in the Internet of Things ( IoT ), cloud and edge computing to help manufacturers advance towards a higher degree of automation.

# About Page

Standard P1935 | About | Standard | System Installation | Forum | Study Group

## ABOUT P1935

Edge computing as an emerging technology that can host the mobile applications closer to its users, provides lower latency, higher efficient bandwidth and service delivery, as well as better user quality of experience. The innovative mobile applications, such as augmented reality, facial detection, and interactive applications, evolve as mobile devices and attract great attention due to their ability to bring convenience and spice up people's lives. With a core concept similar to edge computing of placing the computing capacity at the local area network, fog computing is more often used in Industrial Internet of Things (IIoT) scenarios. Coming up as a modern solution to catch up with such needs, edge/fog computing is a brand-new and promising paradigm to offer an environment characterized by low latency and necessary resources for mobile devices to liberate them from the computing-intensive and real-time applications.

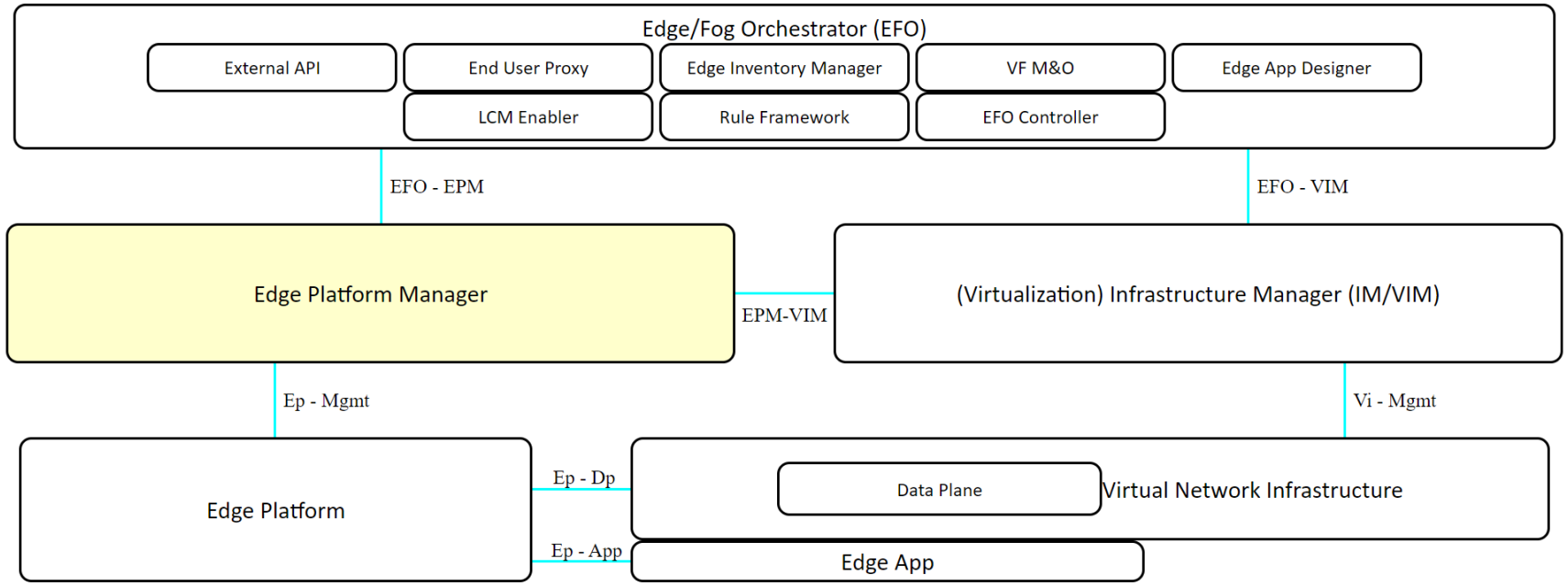
P1935 is a standard that defines the management and orchestration for Edge/Fog Computing, proposed by IEEE EDGEMGBT working group. The standard aims to specify the system archtiecture, necessary opeartions, and relavent APIs for a complete Edge/Fog system.

Aside from the content of the standard, which can be found in the "standard" page, this website also demonstrates an example of implmenetation and gives some guide for the starters.

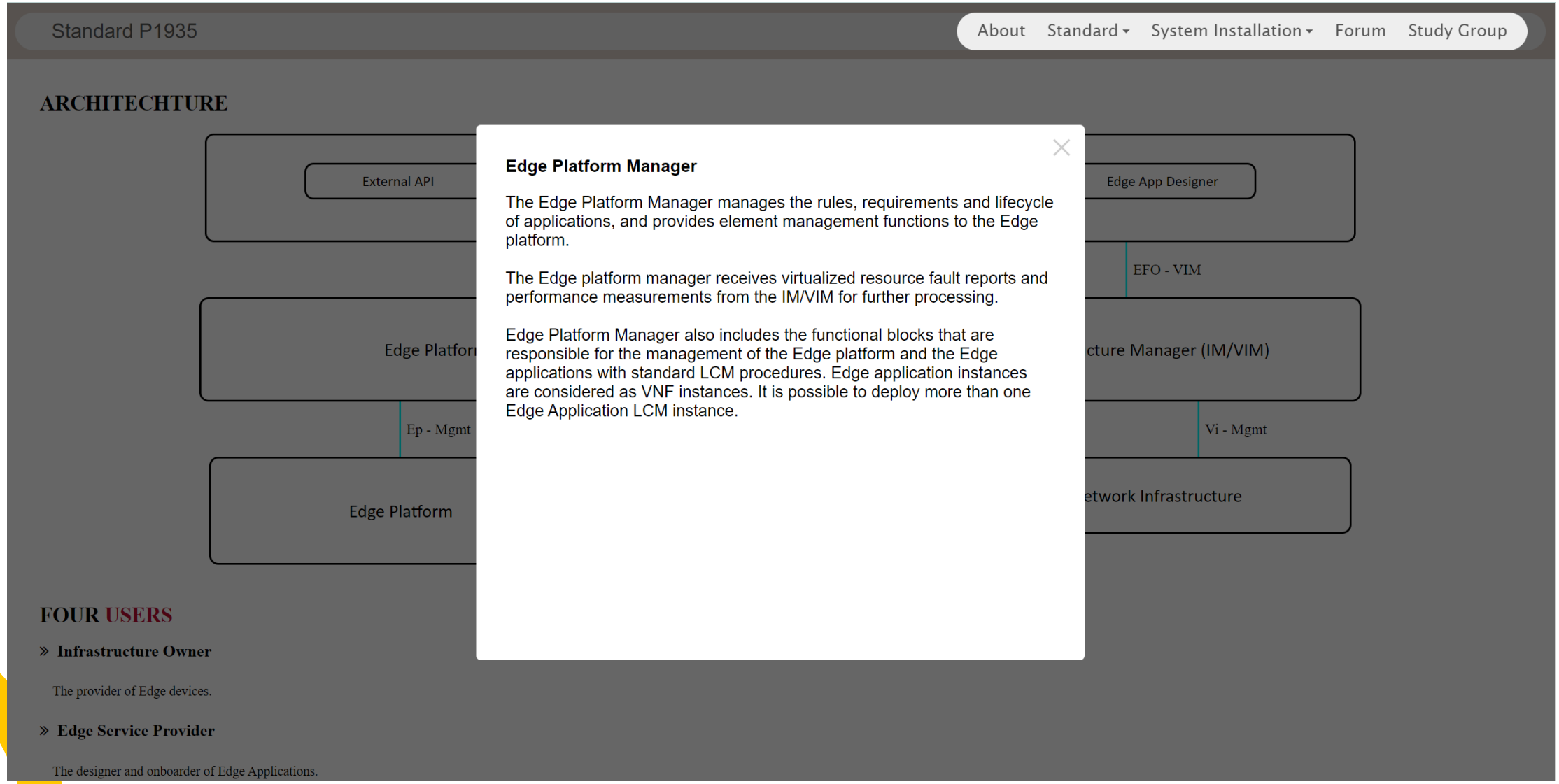
# Standard Page (1/3)

Standard P1935 About Standard System Installation Forum Study Group

**ARCHITECHTURE**



# Standard Page (2/3)



# Standard Page (3/3)

Edge App

## FOUR USERS

» **Infrastructure Owner**

The provider of Edge devices.

» **Edge Service Provider**

The designer and onboarder of Edge Applications.

» **Edge Service Operator**

The service operator dealing with the management and operation of Edge Applications.

» **End User**

The users accessing the system via the End User Apps.

## ACTIONS

» **Resource management**

› Physical machine

› Virtual machine

» **Application management**

› Run in container (Kubernetes)

› Configured by manifest

## TESTBED LOCATION

» **Server IP**

140.112.187.109, 192.168.50.166

» **Path**

/home/1935/standard-p1935

» **Github**

dingyiyi0226/standard-p1935

# System Installation Page

Standard P1935 | About | Standard | System Installation | Forum | Study Group

## SYSTEM INSTALLATION

» **Install step-by-step**

1. Clone the repo

```
git clone https://github.com/dingyiyi0226/standard-p1935.git
```

2. Install python 3.7 with requirements

```
pip install -r requirements.txt
```

3. Install MongoDB

4. Add users in database

```
mongo < init_db.js
```

» **(On a new Ubuntu device) Install by the init script**

› **Edge/Fog Orchestrator**

```
./misc/init.sh efo

The script will
  i. Create a user p1935 with sudo privilege
  ii. Install dependencies
  iii. Install pyenv
  iv. Install mongodb and configure user data
```

› **Control/Compute Node VM Container (Used by creating VM feature on 1935 platform)**

```
./misc/init.sh compute

The script will
  i. Create a user p1935 with sudo privilege
  ii. Install virtualbox, virtualbox-ext-pack
  iii. Download k8s_base.ova
```

## START THE SYSTEM

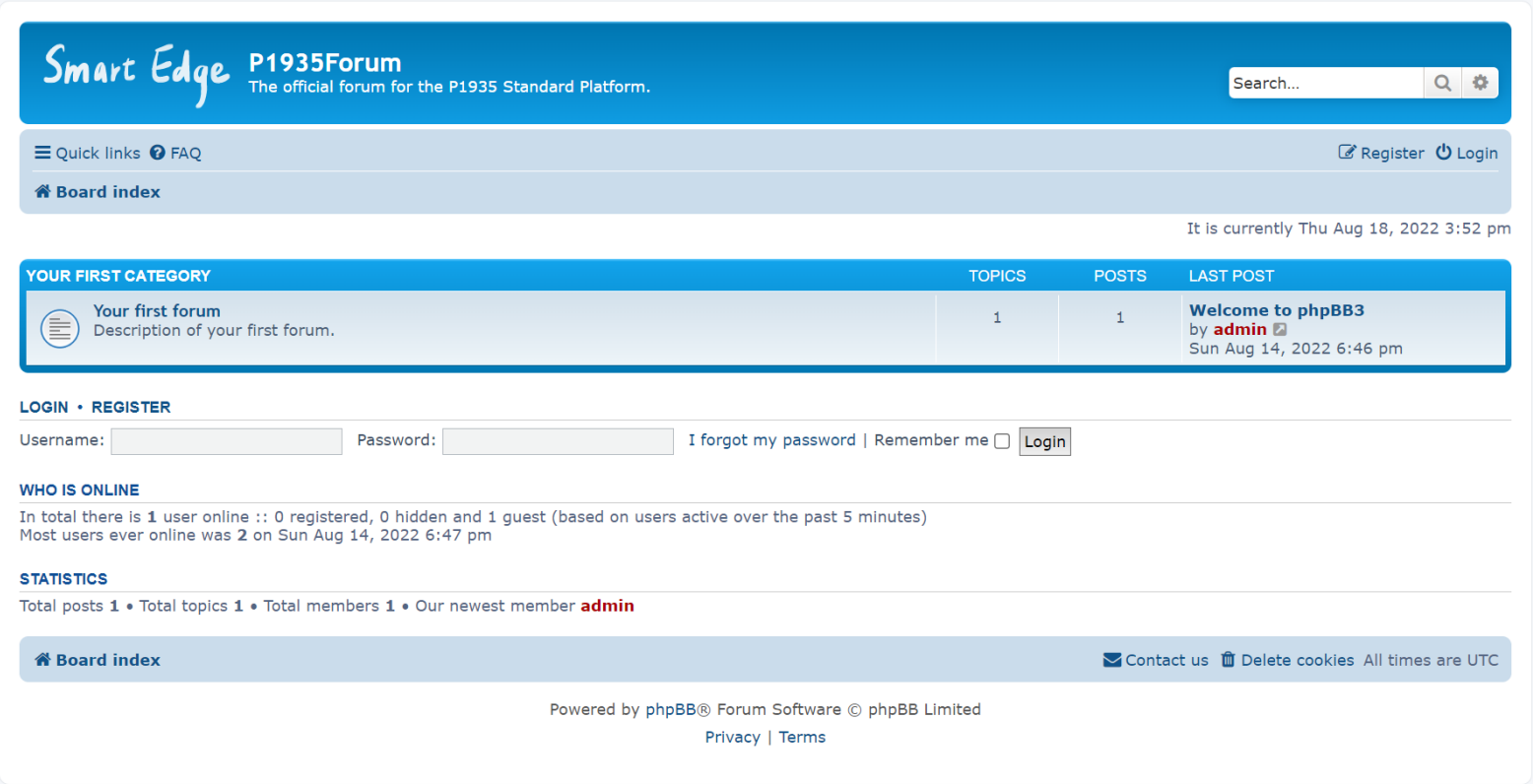
1. (Optional) Switch the user to wmnlab by sudo su - wmnlab (Required if you need to create vm)

2. Activate the python environment by conda activate 1935env

3. Start Edge/Fog Orchestrator by executing ./run_dev.sh or ./run_prod.sh from develop/production mode.

4. Go to http://192.168.50.166:5000/ in web browser to access the UI of Edge/Fog Orchestrator.

# Forum Page


Smart Edge
P1935Forum
The official forum for the P1935 Standard Platform.
Search...
Quick links
FAQ
Register
Login
Board index
It is currently Thu Aug 18, 2022 3:52 pm
YOUR FIRST CATEGORY
TOPICS
POSTS
LAST POST
Your first forum
Description of your first forum.
1
1
Welcome to phpBB3
by admin
Sun Aug 14, 2022 6:46 pm
LOGIN • REGISTER
Username:
Password:
I forgot my password | Remember me
Login
WHO IS ONLINE
In total there is 1 user online :: 0 registered, 0 hidden and 1 guest (based on users active over the past 5 minutes)
Most users ever online was 2 on Sun Aug 14, 2022 6:47 pm
STATISTICS
Total posts 1 • Total topics 1 • Total members 1 • Our newest member admin
Board index
Contact us
Delete cookies
All times are UTC
Powered by phpBB® Forum Software © phpBB Limited
Privacy | Terms

# Study Group Page

Standard P1935

About | Standard | System Installation | Forum | Study Group

| Date | Topic | Replay | Pdf |
|---|---|---|---|
| **Feb 22nd, 2022** | IoT Management Platform based on MQTT | | |
| **Mar 9th, 2022** | Paper review Mobile Edge Computing A key technology towards 5G | | |
| **Apr 13th, 2022** | Introduction and Implementation of standard P1935 | | |
| **May 4th, 2022** | SiMPLE Survivability in MultiPath Link Embedding | | |
| **May 11th, 2022** | Open5GS Deployment on 1935 tutorial | | |
| **May 23rd, 2022** | The Guidance to Become an O-RAN Engineer | | |

# Use Case Demo

Presenter: 吳建翰

# Introduction

1. P1935 standard enables orchestration for application and resources. Applications like object recognition, facial recognition and video streaming can be implemented on the platform [1].

2. In this research, we are going to simulate the connection of gNodeB (gNB) and 5G Core Network (Fig 2).

3. Because unexpected crash may occur on both gNB and any core network function (NF), we design an algorithm to rebuild the connection and keep the status simultaneously.

4. We first introduce the mechanism and implementation of both stateful and stateless services. We make gNB stateless and AMF stateful. Besides, we design an algorithm to implement the reconnection mechanism.

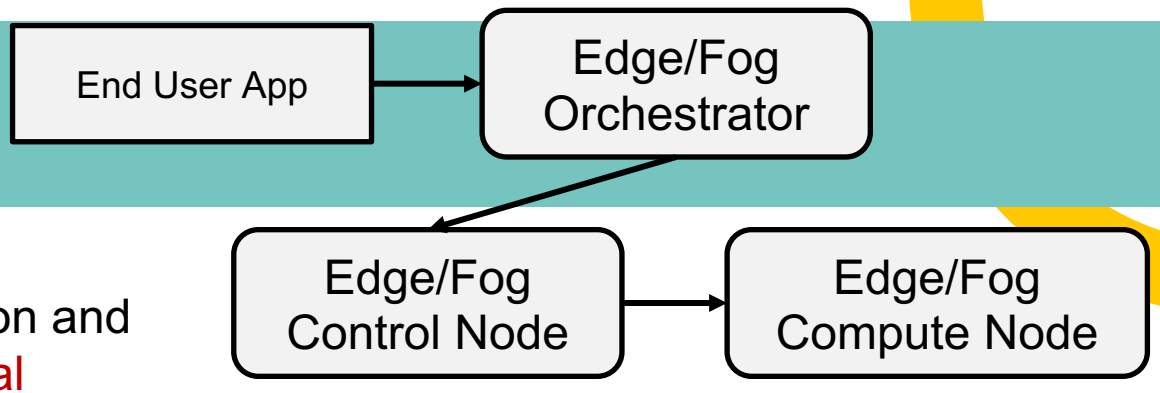



Figure 1. The basic architecture of the Edge/Fog system framework in P1935 standard

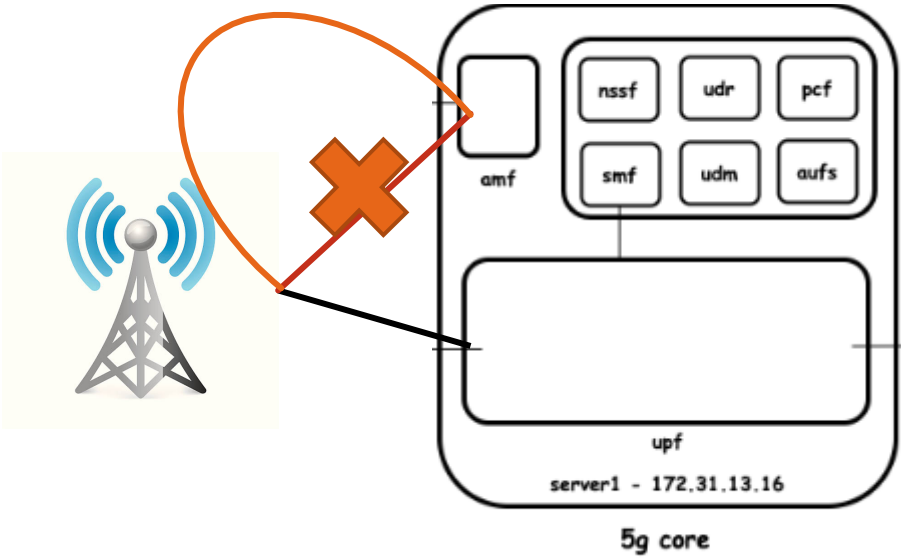
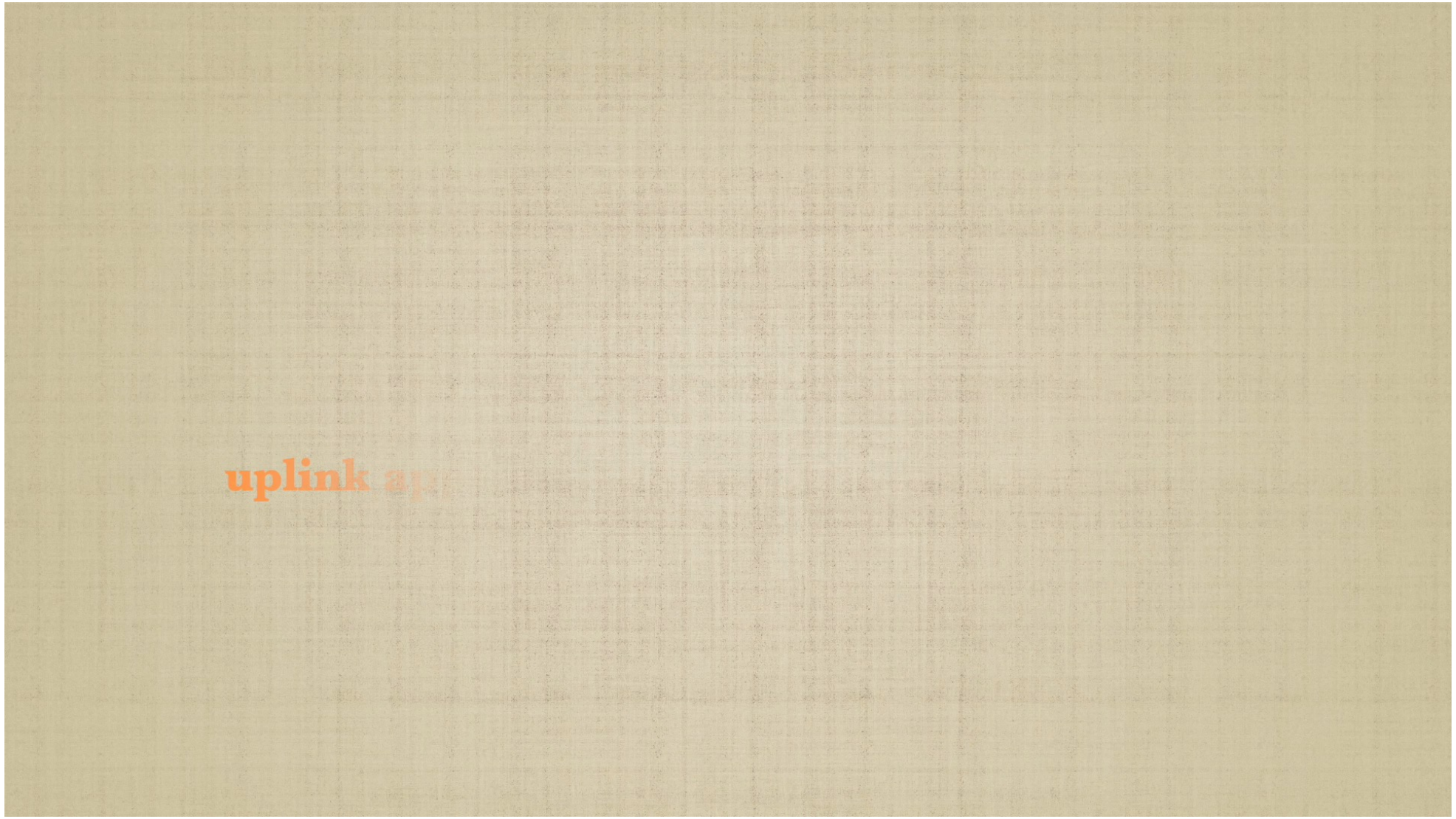



Figure 2. Connection between gNB and the Core Network.

[1] 5G Edge Computing Experiments with Intelligent Resource Allocation for Multi-Application Video Analytics

# Previous Work Demo - Uplink



# Onboarding Demo

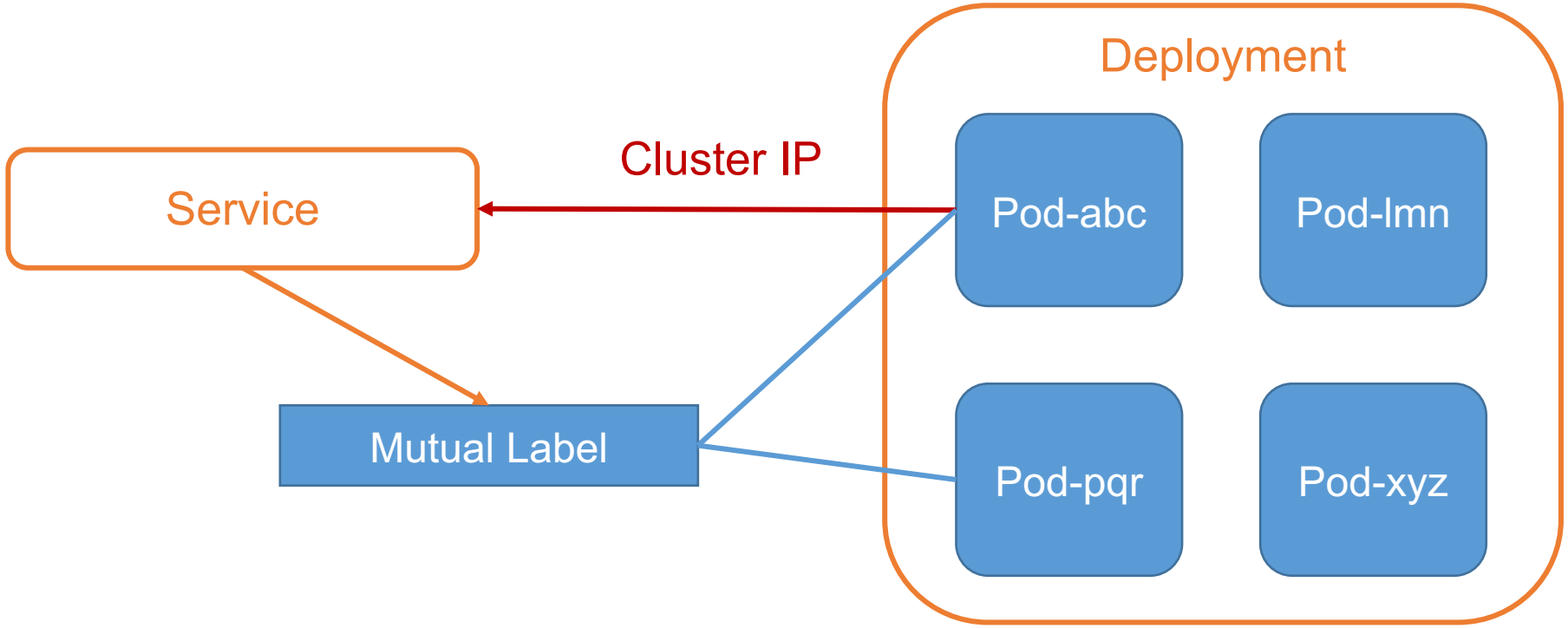
State-less App

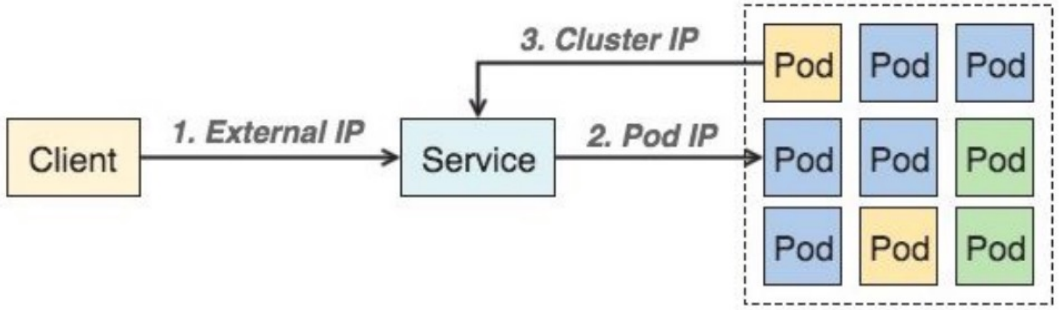
- Data is deleted when a pod terminates.
- We use UE-RAN-Sim gNb as an example.
- *UE-RAN-Sim* simulates network simulation for 5G NR RAN and UE.

State-ful App

- *StatefulSet* manages the deployment and scaling, and provides guarantees about the ordering and uniqueness of pods.
- We deploy stateful *Open5GS* as a core network.

# Stateless Application

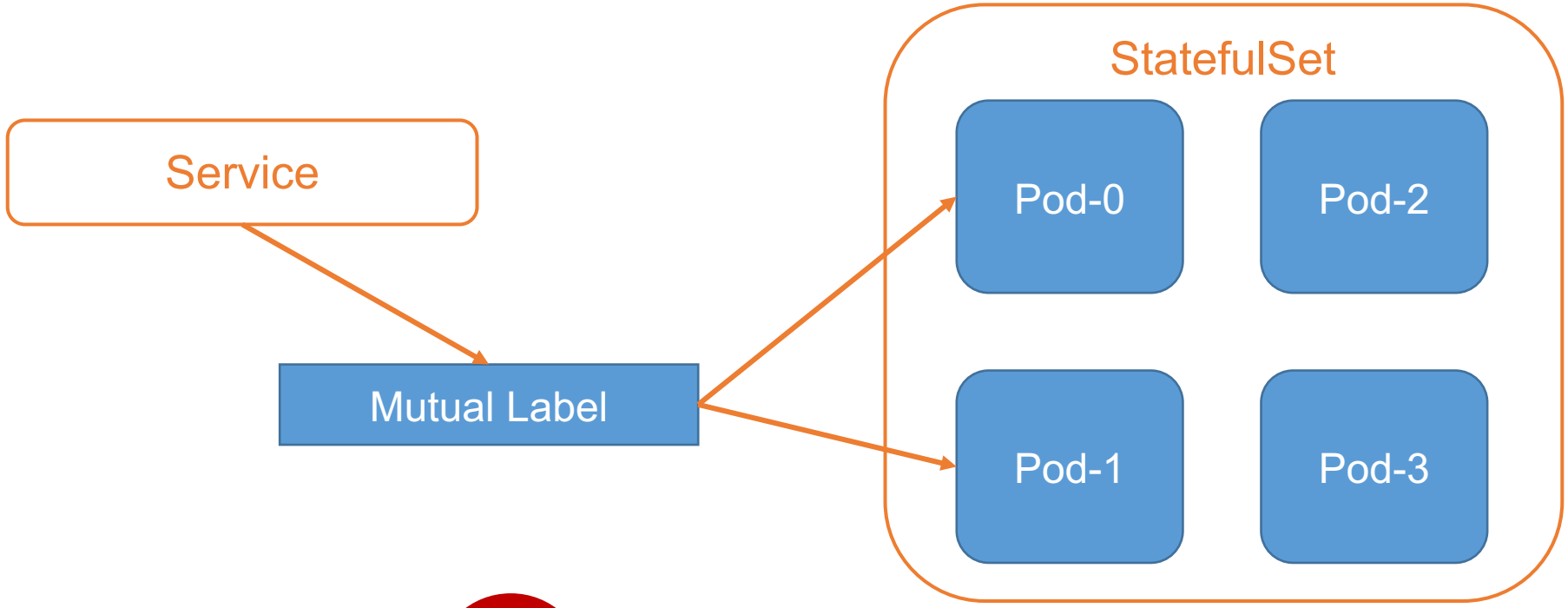


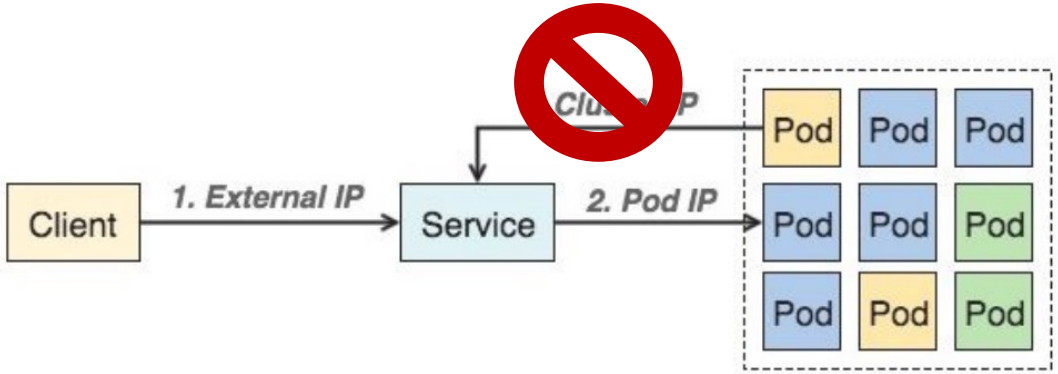




Replicas in deployment

```
nginx-deployment-66b6c48dd5-fr892  172.16.209.242
nginx-deployment-66b6c48dd5-m6nkw  172.16.209.240
nginx-deployment-66b6c48dd5-mv4qj  172.16.209.241
```

# Stateful Application



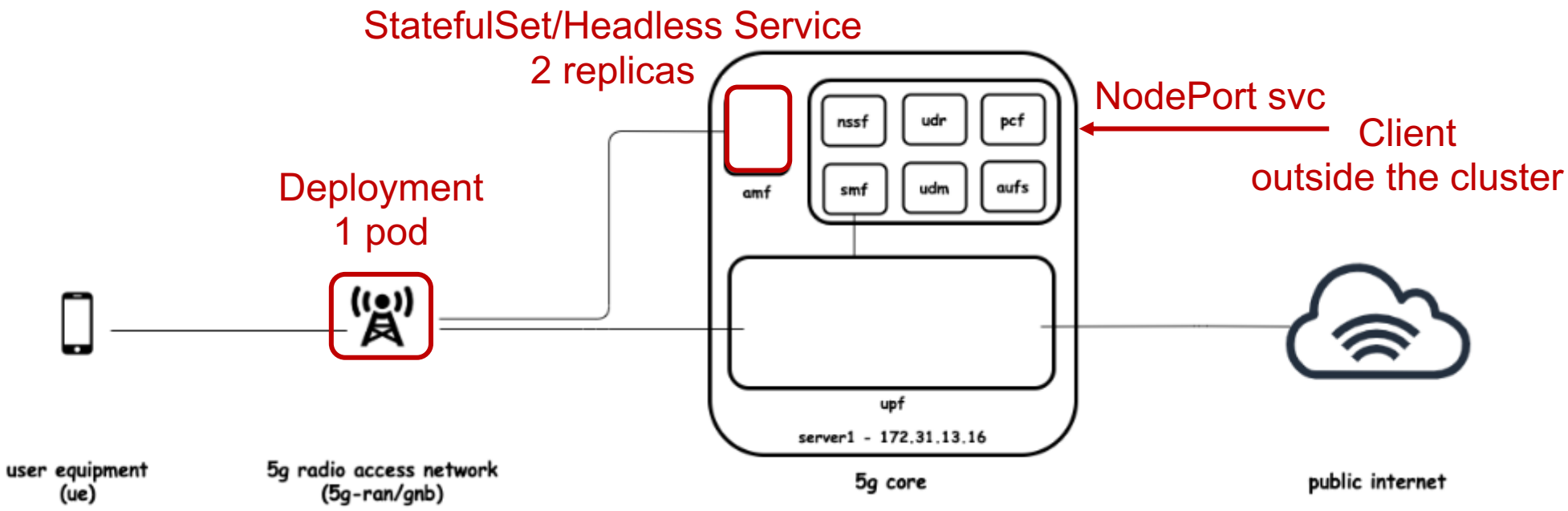




Replicas in StatefulSet

```
open5gs-amf-0   172.16.209.236
open5gs-amf-1   172.16.209.230
```

Headless service to return Pod IP addresses

# Communication Architecture



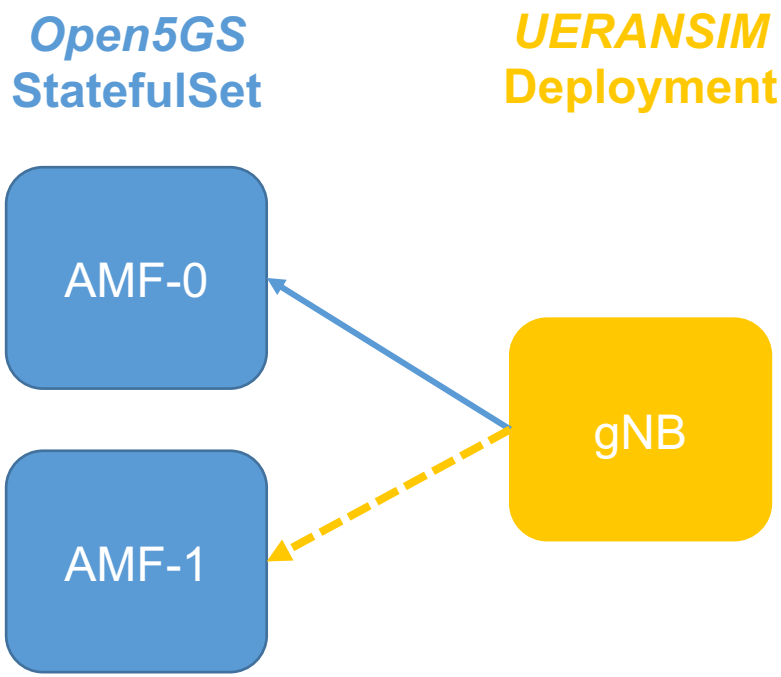

### *UERANSIM*

- Stateless Deployment to manage 1 gNB pod.
- gNB binds node IP and AMF pod IP.

### *Open5GS*

- StatefulSet to manage pods.
- We use AMF as the testcase with 2 replicas.

# Proposed Approach

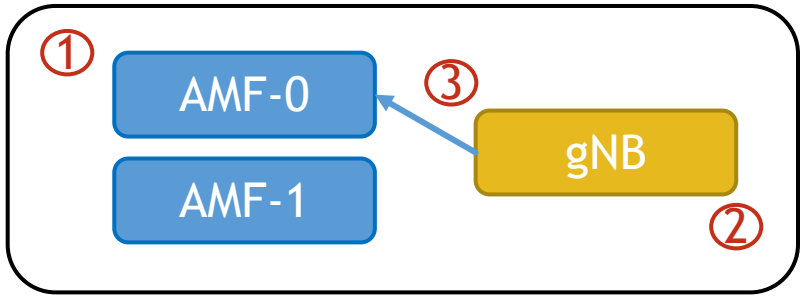


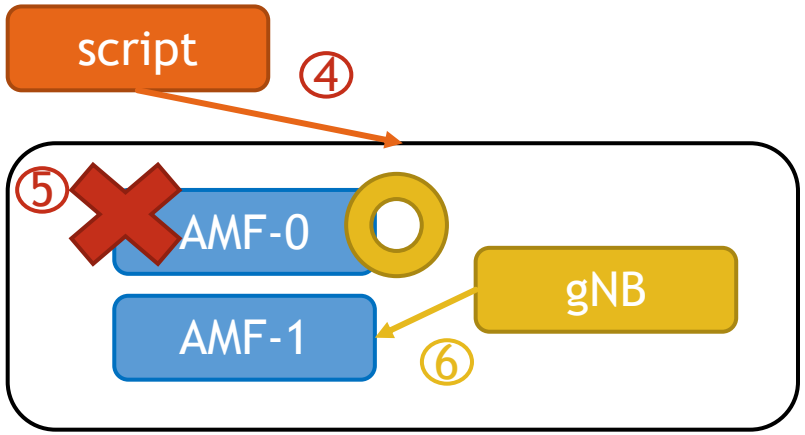

```
set serving_index to 0 // use amf-0 to link gNB
set amf-0 IP into gNB config file
deploy gNB to platform

while (true) {
    if (serving_index == 0 and amf-0 crashed) {
        set serving_index to 1
        set amf-1 IP into gNB config file
        restart gNB container // very fast
        delete amf-0 and wait amf-0 restarted
        get amf-0 IP
    } else if (serving_index == 1 and amf-1 crashed) {
        set serving_index to 0
        set amf-0 IP into gNB config file
        restart gNB container // very fast
        delete amf-1 and wait amf-1 restarted
        get amf-1 IP
    }
}
```

# Demo Video

Demo Flow:

1. Deploy 5G Core Network as StatefulSet services. Among them, the AMF service contains 2 replicas.
2. Deploy UERANSIM gNB as stateless services (we use deployment only).
3. Make sure the network connection is successful. Namely, gNB is connected to one of the AMF pods.
4. Run the detection script and watch core network pods.
5. Delete the connected AMF pod and the script should detect the error.
6. Check whether the gNB is connected to another AMF pod as soon as possible. Besides, check whether the deleted AMF pod is restarted.



ubuntu@ubuntu1804: ~/open

```
statefulset.apps "open5gs-pcf" deleted
configmap "open5gs-smf-config" deleted
service "open5gs-smf-svc-pool" deleted
statefulset.apps "open5gs-smf" deleted
configmap "open5gs-smf-diameter" deleted
configmap "open5gs-udm-config" deleted
service "open5gs-udm-svc-pool" deleted
statefulset.apps "open5gs-udm" deleted
configmap "open5gs-udr-config" deleted
service "open5gs-udr-svc-pool" deleted
statefulset.apps "open5gs-udr" deleted
configmap "open5gs-upf-config" deleted
service "open5gs-upf-svc-pool" deleted
statefulset.apps "open5gs-upf" deleted
service "open5gs-webui" deleted
deployment.apps "open5gs-webui" deleted
namespace "open5gs" deleted
ubuntu@ubuntu1804:~/open5gs_k8s/open5gs_sa$ ./undeploy.sh
```

ubuntu@ubuntu1804: ~

```
ubuntu@ubuntu1804:~$
```

ubuntu@ubuntu1804: ~

```
Every 2.0s: kubectl get pods -n open5gs ubuntu1804: Fri Aug 19 11:09:06 2022

No resources found in open5gs namespace.
```

ubuntu@ubuntu1804: ~

```
ubuntu@ubuntu1804:~$
```

11:09 PM
8/19/2022

# Caching Related Work

Presenter: 陳巧錚

# Outline

1 Introduction

2 System Architecture

3 P1935 APIs

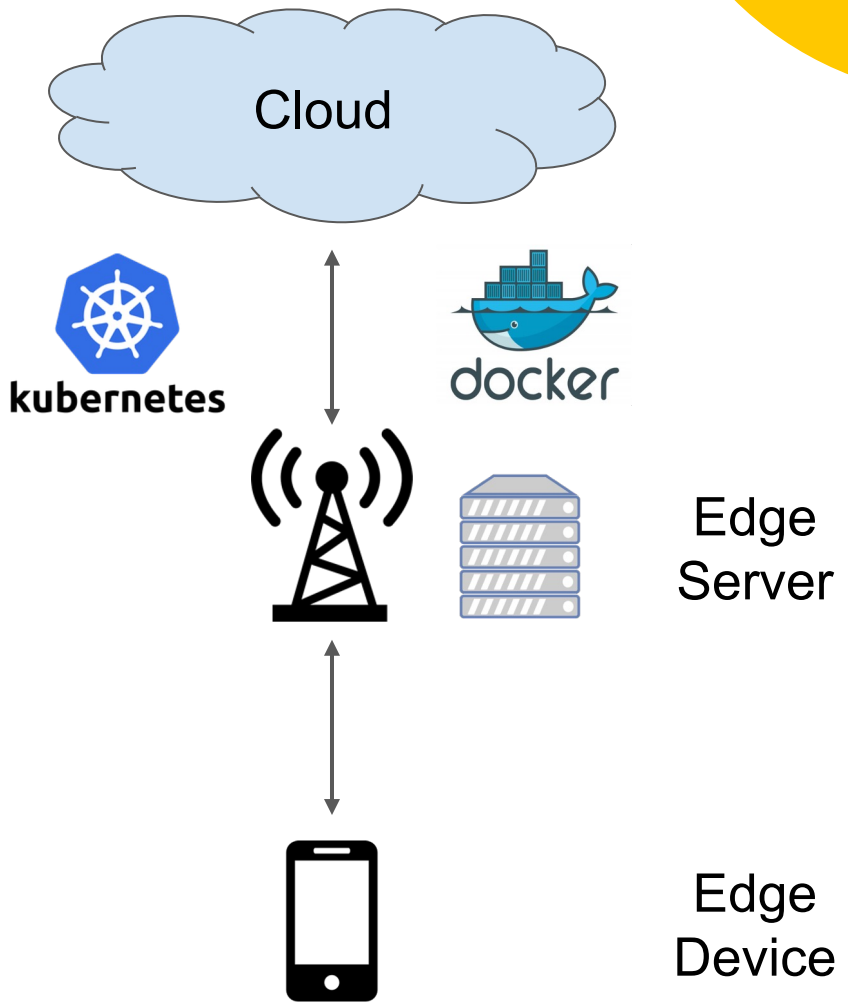
- Resource Management
- Application Management

4 Future Work

# 01 Introduction

# Background

- Mobile/Multi-access Edge Computing (MEC)
  - Reduce latency
    - Improved user experience
  - Virtualization
    - e.g. container & Kubernetes



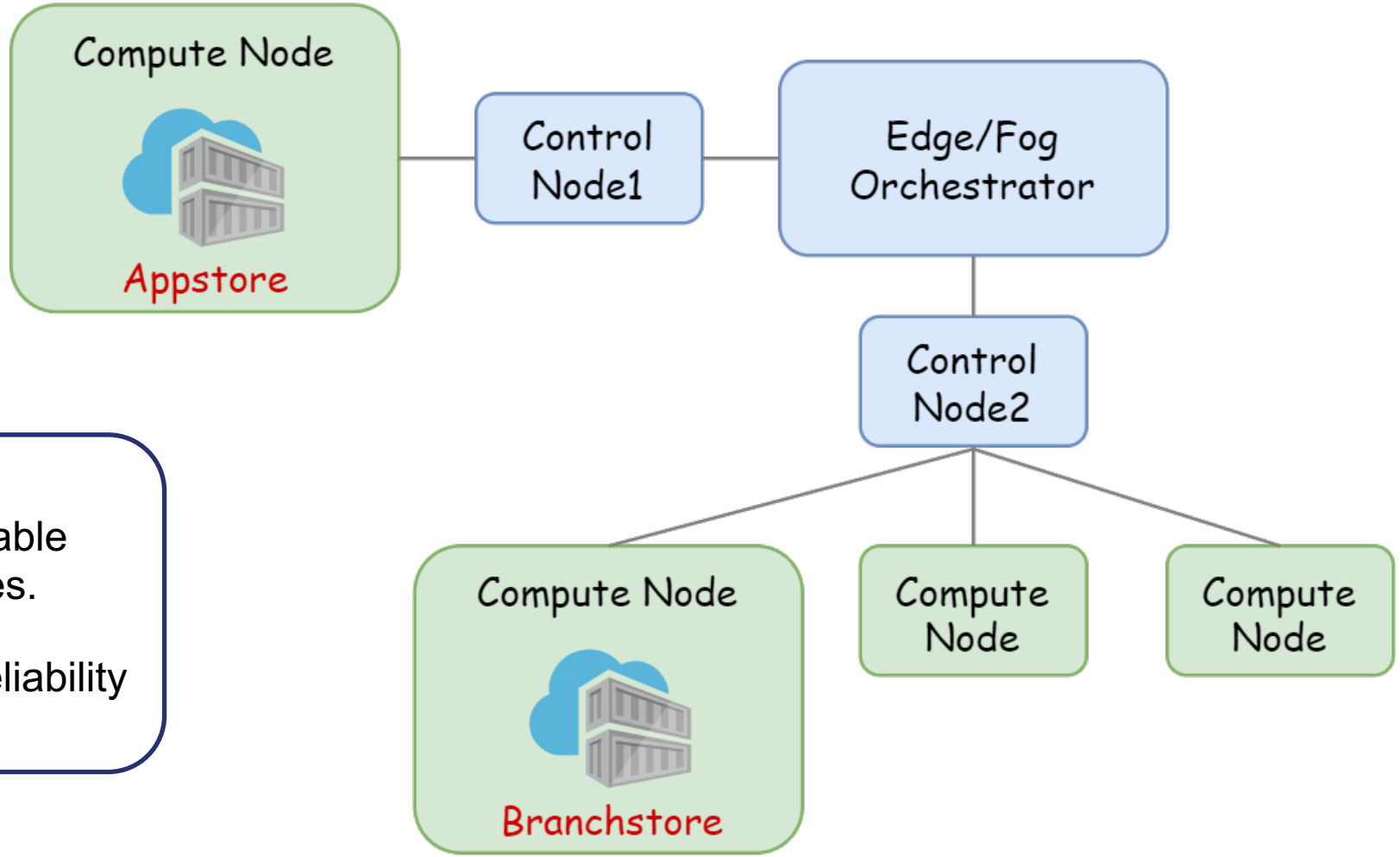
# Challenges

- Long container startup latency
  - Limited edge resources
  - High round-trip time (RTT)
- Bloated image size
  - Complex software dependencies e.g. TensorFlow
  - ML/AI based tasks
- Frequent deployment
  - Application update
  - User mobility

02

# System Architecture

# Architecture



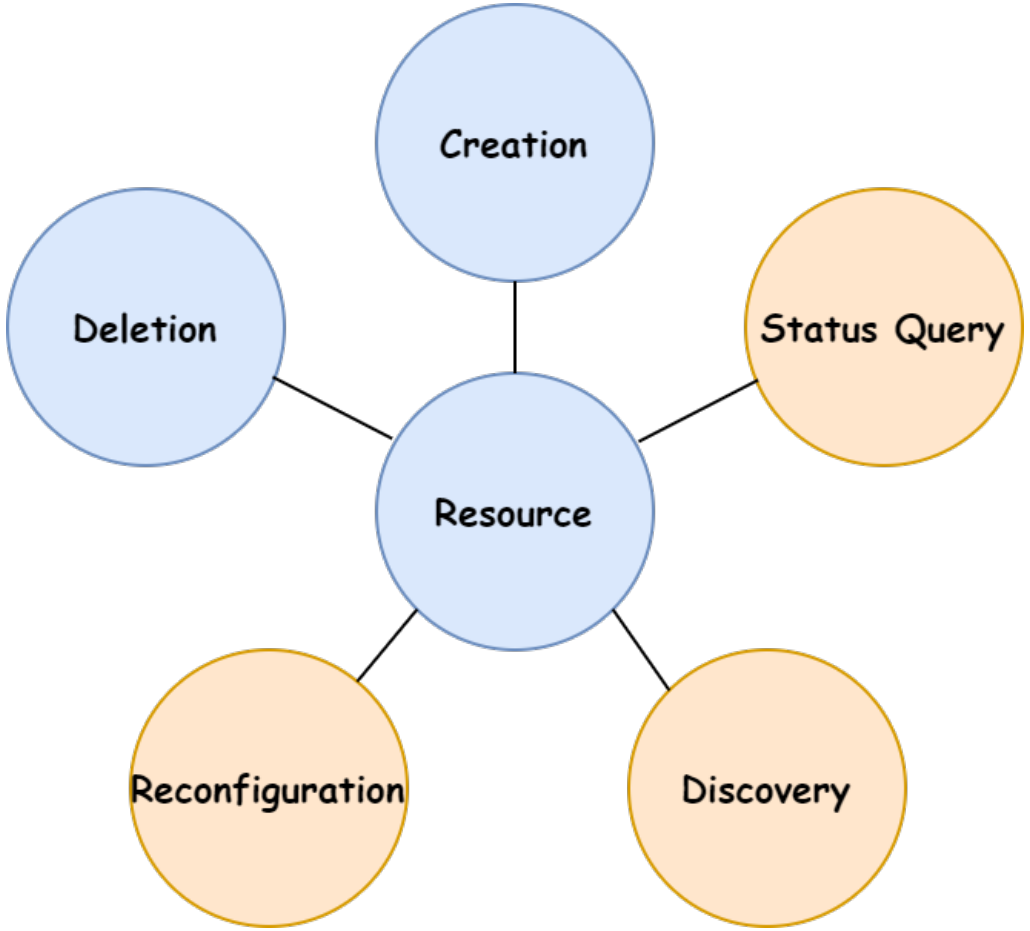

- Dynamically deploy suitable numbers of branch stores.
- Improve flexibility and reliability

03

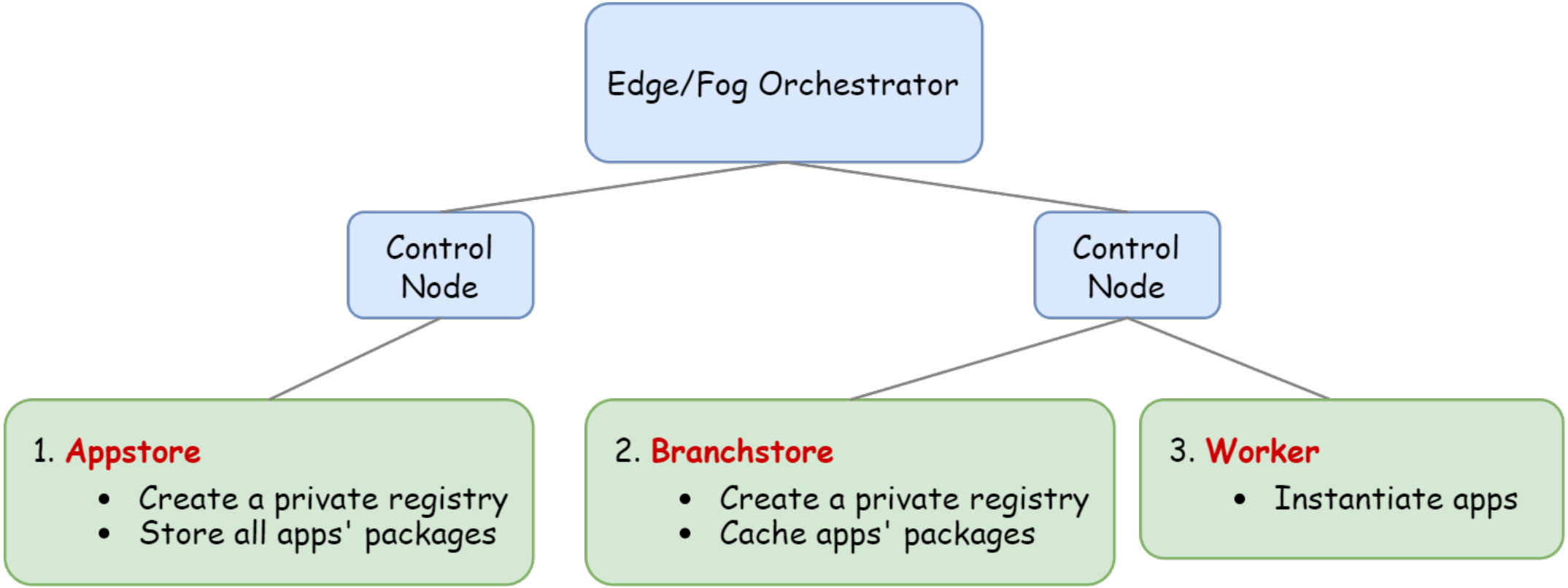
# P1935 APIs

- Resource Management
- Application Management

# Resource Management



# Roles

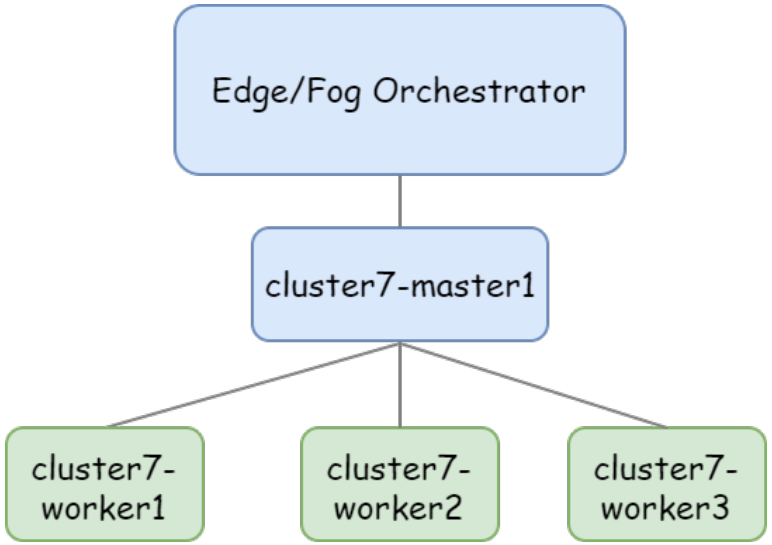



Control nodes

Compute nodes

# Yaml File

```yaml
spec:
  imagePullSecrets:
    - name: appstore
  containers:
    - name: demoapp
      imagePullPolicy: IfNotPresent
      image: 192.168.50.12:5000/demoapp
      ports:
      - containerPort: 5678
  nodeSelector:
    my_role: "worker"
```




```
ubuntu@ubuntu1804:~$ kubectl get nodes --show-labels
NAME              STATUS   ROLES                  AGE   VERSION   LABELS
cluster7-master1  Ready    control-plane,master   28d   v1.20.5   beta.kubernetes.io/arch=amd64
,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=cluster7-master1,ku
bernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node-role.kubernetes.io/master=
cluster7-worker1  Ready    <none>                 28d   v1.20.5   beta.kubernetes.io/arch=amd64
,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=cluster7-worker1,ku
bernetes.io/os=linux,my_role=branchstore
cluster7-worker2  Ready    <none>                 28d   v1.20.5   beta.kubernetes.io/arch=amd64
,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=cluster7-worker2,ku
bernetes.io/os=linux,my_role=worker
cluster7-worker3  Ready    <none>                 17m   v1.20.5   beta.kubernetes.io/arch=amd64
,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=cluster7-worker3,ku
bernetes.io/os=linux,my_role=worker
```

# Resource Status Query

## get_cache_info

Get caching information of a branchstore or an appstore.

### Request

```
GET /api/nodes/<nodeName>/cache
```

Example:

```
{
  "image_list": [
    "demoapp"
  ],
  "role": "branchstore",
  "size": "93.9 MB"
}
```

### Request parameters

| Parameter name | Value | Description |
|---|---|---|
| `nodeName` | `string` | node name |

### Response

If successful, this method returns caching information with three fields: `role`, `image_list`, `size`

### Response details

| Status | Description |
|---|---|
| **200** OK | |
| **400** Bad Request | Action failed |
| **403** Forbidden | User is not authorized |

# Resource Discovery

1 ) ## list_branchstore

List all branchstores

### Request

```
GET /api/branchstore
```

2 ) ## list_appstore

List the appstore

### Request

```
GET /api/appstore
```

### Response

If successful, this method returns a list of branchstores in the response body /appstores

#### Response details

| Status | Description |
|---|---|
| **200** OK | |
| **400** Bad Request | No branstore /appstore |
| **403** Forbidden | User is not authorized |

Example:


```
[
  {
    "cache": [
      "demoapp"
    ],
    "capacity": "1 GB",
    "current_size": "93.9 MB",
    "ip": "192.168.50.31",
    "name": "cluster7-worker1"
  }
]
```

# Resource Reconfiguration (1/3)

1 ) create_branchstore

Change a worker to a branchstore.

Request

```
PATCH /api/cluster/<clusterName>/nodes/<nodeName>/create_branchstore
```

2 ) create_appstore

Change a worker to an appstore.

Request

```
PATCH /api/cluster/<clusterName>/nodes/<nodeName>/create_appstore
```

Request parameters

| Parameter name | Value | Description |
|---|---|---|
| `clusterName` | `string` | cluster name |
| `nodeName` | `string` | node name |

Response

If successful, this method returns `branchstore's name` in the response body /appstore's

Response details

| Status | Description |
|---|---|
| **200** OK | |
| **400** Bad Request | Action failed |
| **403** Forbidden | User is not authorized |

- Setup Docker private registry

1. Set username and password

```
$ sudo apt-get install apache2-utils
$ htpasswd –Bbn <username> <password> > auth/htpasswd
```

2. Start the registry container

```
$ docker run -d -p 5000:5000 --restart=always --name registry -v /reg:/var/lib/registry -
v `pwd`/auth:/auth -e "REGISTRY_AUTH=htpasswd" -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry
Realm" -e "REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd" registry:2
```

3. Enable REGISTRY_STORAGE_DELETE_ENABLED=true

4. Tag and push image to the registry

```
$ docker pull mysql
$ docker tag mysql registry_ip:5000/mysql
$ docker push registry_ip:5000/mysql
```

5. Get stored image

```
$ curl --user <username>:<password> http://registry_ip:5000/v2/_catalog
```

```
(1935env) chiao@wmnlab-server:~$ curl --user ubuntu:ubuntu http://192.168.50.42:5000/v2/_catalog
{"repositories":["demoapp","mysql","nginx","ubuntu2004"]}
```

# Resource Reconfiguration (2/3)

## 3 ) remove_branchstore

Change a branchstore to a worker.

### Request

```
PATCH /api/cluster/<clusterName>/nodes/<nodeName>/remove_branchstore
```

## 4 ) remove_appstore

Change an appstore to a worker.

### Request

```
PATCH /api/cluster/<clusterName>/nodes/<nodeName>/remove_appstore
```

### Request parameters

| Parameter name | Value | Description |
|---|---|---|
| `clusterName` | `string` | cluster name |
| `nodeName` | `string` | node name |

### Response

If successful, this method returns `branchstore's name` in the response body /appstore's

### Response details

| Status | Description |
|---|---|
| **200** OK | |
| **400** Bad Request | Action failed |
| **403** Forbidden | User is not authorized |

# Resource Reconfiguration (3/3)

## 5 ) apply_cache

Cache app image to the branchstore.

### Request

```
PATCH /api/nodes/<nodeName>/apply_cache/<appImage>
```

## 6 ) remove_cache

Remove cached app image from the branchstore/appstore.

### Request

```
PATCH /api/nodes/<nodeName>/remove_cache/<appImage>
```

Request parameters

| Parameter name | Value | Description |
|---|---|---|
| `nodeName` | `string` | node name |
| `appImage` | `string` | app image name |

### Response

If successful, this method returns `appImage` in the response body

Response details

| Status | Description |
|---|---|
| **200** OK | |
| **400** Bad Request | Action failed |
| **403** Forbidden | User is not authorized |

- Remove images from the private registry

1. Get the Docker-Content-Digest of the tag

```
$ curl --user <username>:<password> -s -v http://registry_ip:5000/v2/<image_name>/
manifests/latest -H 'Accept: application/vnd.docker.distribution.manifest.v2+json'
```

```
< HTTP/1.1 200 OK
< Content-Length: 2618
< Content-Type: application/vnd.docker.distribution.manifest.v2+json
< Docker-Content-Digest: sha256:505290a5af407b56452715c6128ba7da8370786fc1559e6b4c
< Docker-Distribution-Api-Version: registry/2.0
< Etag: "sha256:505290a5af407b56452715c6128ba7da8370786fc1559e6b4c8d0e6299293b38"
< X-Content-Type-Options: nosniff
```

2. Delete the image using API

```
$ curl --user <username>:<password> -s -X DELETE http://
registry_ip :5000/v2/hello-world/manifests/<Docker-Content-Digest_value> -v
```

3. Run the garbage collector

```
$ docker exec -it registry bin/registry garbage-collect --delete-untagged
/etc/docker/registry/config.yml
```

4. Restart the registry

# Application Management

## 1 ) worker_instantiation

Instantiate the app (caching version)

### Request

```
GET /api/apps/<appID>/worker_instantiate
```

Request parameters

| Parameter name | Value | Description |
|---|---|---|
| `appID` | `string` | app ID |

### Response

If successful, this method returns `appID` in the response body

Response details

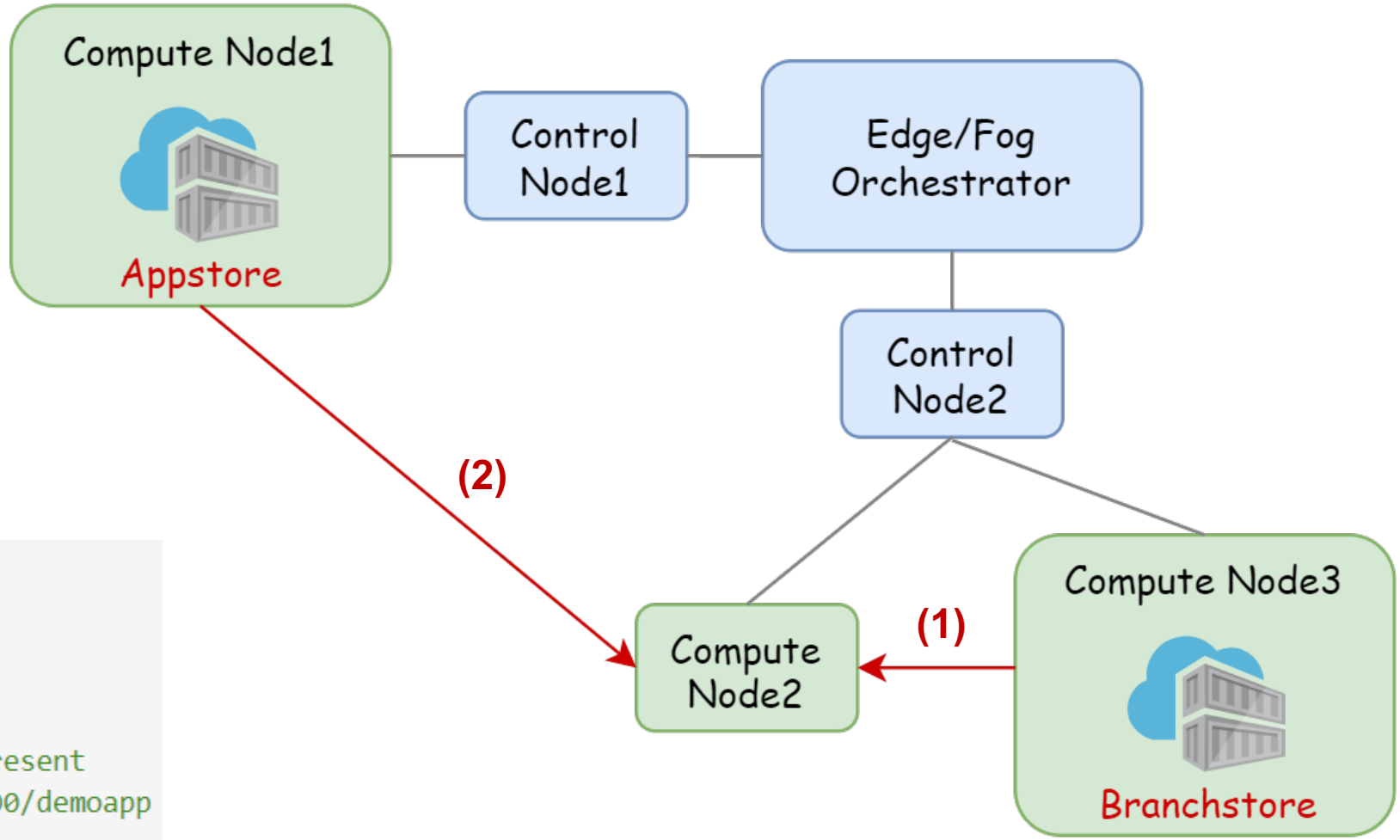
| Status | Description |
|---|---|
| **200** OK | |
| **400** Bad Request | Action failed |
| **403** Forbidden | User is not authorized |

## 2 ) terminate_app

Terminate the app

### Request

```
GET /api/apps/<appID>/terminate
```




yaml file:

```yaml
spec:
  imagePullSecrets:
    - name: appstore
  containers:
    - name: demoapp
      imagePullPolicy: IfNotPresent
      image: 192.168.50.12:5000/demoapp
      ports:
      - containerPort: 5678
  nodeSelector:
    my_role: "worker"
```

# 04 Future Work

# Future Work

- Current:
  - Create caching related APIs in p1935 platform.
  - Implement caching algorithm LRU.

- Future Work:
  - Design caching algorithm.
  - Improve the procedure of downloading application images.

# Create a over WAN cluster and Traffic Forwarding

Presenter: 賈承叡

# Introduction

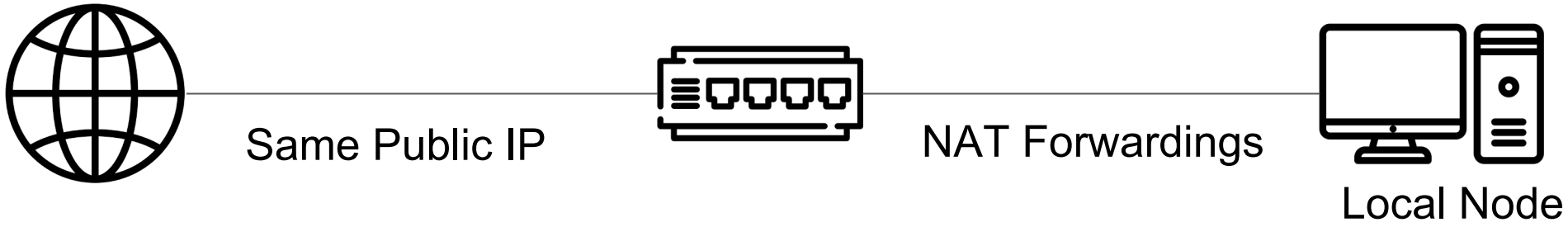
- Create a Over WAN Cluster
- Forwarding policy and load balancer on Kubernetes

# Create a Over WAN Cluster

# Network Architecture

- Before init a Cluster


- After init a Cluster



# Before Init a Cluster (1/2)

- Check if there is a network interface that has public IP at local node.

```
$ ip a

enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
  link/ether 08:00:27:7e:61:e1 brd ff:ff:ff:ff:ff:ff
  inet 140.112.XXX.XXX/24 brd 140.112.XXX.XXX scope global dynamic enp0s3
  valid_lft 82258sec preferred_lft 82258sec
```

- Check the firewall configuration (Example for defaults)

| Protocol | Direction | Ports |
|---|---|---|
| TCP | Incoming | 2379,2380,6443,10250,10257,10259 |
| UDP | Bidirectional | 4789 |
| TCP | Incoming | 30000-32767 (NodePort Services) |

# Before Init a Cluster (2/2)

- Modify the Container Network Interface (Calico)
  - Use VXLAN for all cloud VM capability (Link)
  - Original File: https://projectcalico.docs.tigera.io/manifests/calico.yaml

```yaml
- name: CALICO_IPV4POOL_IPIP
  value: "Never"
...
- name: CALICO_IPV4POOL_VXLAN
  value: "Always"
```

```yaml
livenessProbe:
    exec:
      command:
      - /bin/calico-node
      - -felix-live
     # - -bird-live
readinessProbe:
    exec:
      command:
      - /bin/calico-node
      - -felix-ready
      # - -bird-ready
```

# Init a cluster using Kubeadm (1/2)

- Make sure the commands are using public IP :
  - --control-plane-endpoint
  - --apiserver-advertise-address

```
$ sudo kubeadm init --pod-network-cidr=172.16.0.0/16
\ --control-plane-endpoint=140.112.xxx.xxx
\ --apiserver-advertise-address=140.112.xxx.xxx

$ mkdir -p $HOME/.kube
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

# Init a cluster using Kubeadm (2/2)

- Apply the modified “calico.yaml”

```
$ kubectl apply -f calico.yaml
```

- Join the cluster

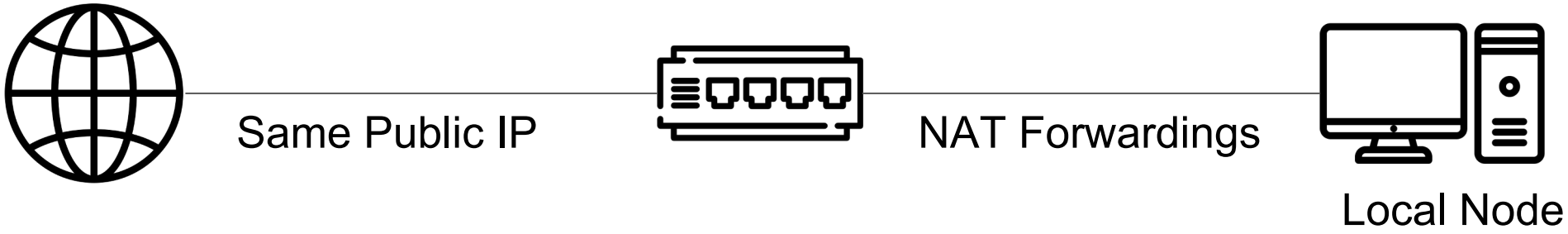
```
$ kubeadm join 140.112.XXX.XXX:6443 --token ... --discovery-token-ca-cert-hash sha256:...
```

- Change the network architecture and wait for reconnection.

# Network Architecture

- Before Init a Cluster


- After Init a Cluster



# After Create a Cluster (1/2)

instance-1 – Compute Engine | ssh.cloud.google.com/v2/ssh/proje

https://ssh.cloud.google.com/v2/ssh/projects/fair-catcher-34461

直接透過瀏覽器進行 SSH 連線

```
billy_12543@instance-1:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:04:29:64:e7  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens4: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1460
        inet 10.128.0.2  netmask 255.255.255.255  broadcast 0.0.0.0
        inet6 fe80::4001:aff:fe80:2  prefixlen 64  scopeid 0x20<link>
        ether 42:01:0a:80:00:02  txqueuelen 1000  (Ethernet)
        RX packets 4288  bytes 2068987 (2.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4136  bytes 532867 (532.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2654  bytes 331390 (331.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2654  bytes 331390 (331.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vxlan.calico: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1410
        inet 172.16.23.128  netmask 255.255.255.255  broadcast 0.0.0.0
        inet6 fe80::6468:4aff:fef0:b376  prefixlen 64  scopeid 0x20<link>
        ether 66:68:4a:f0:b3:76  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 13 overruns 0  carrier 0  collisions 0

billy_12543@instance-1:~$
```

Kubernetes Dashboard

https://192.168.31.237:30950/#/node/instance-1?namespace=_all

kubernetes | 全部命名空間 | 搜索

Cluster > Nodes > instance-1

- Replica Sets
- Replication Controllers
- Stateful Sets
- **Service** N
  - Ingresses
  - Services
- **Config and Storage**
  - Config Maps N
  - Persistent Volume Claims N
  - Secrets N
  - Storage Classes
- **Cluster**
  - Cluster Role Bindings
  - Cluster Roles
  - Events N
  - Namespaces
  - Network Policies N
  - Nodes
  - Persistent Volumes
  - Role Bindings N
  - Roles N

## 條件

| 類別 | 狀態 | 最後的檢查時間 | 最後的遷移時間 |
|---|---|---|---|
| NetworkUnavailable | False | 19 minutes ago | 19 minutes ago |
| MemoryPressure | False | 20 seconds ago | 19 minutes ago |
| DiskPressure | False | 20 seconds ago | 19 minutes ago |
| PIDPressure | False | 20 seconds ago | 19 minutes ago |
| Ready | True | 20 seconds ago | 19 minutes ago |

## Pods

| 名字 | 命名空間 | 鏡像 | 標籤 | 節點 | 狀態 |
|---|---|---|---|---|---|
| calico-node-hmlpt | kube-system | docker.io/calico/node:v3.23.3 | controller-revision-hash: 877679cf<br>k8s-app: calico-node<br>pod-template-generation: 2 | instance-1 | Running |
| kube-proxy-ghsdj | kube-system | k8s.gcr.io/kube-proxy:v1.24.3 | controller-revision-hash: 94985b49<br>k8s-app: kube-proxy<br>pod-template-generation: 1 | instance-1 | Running |

# After Create a Cluster (2/2)

- At local node:

```
$ ip a

enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
  link/ether 08:00:27:7e:61:e1 brd ff:ff:ff:ff:ff:ff
  inet 192.168.XXX.XXX/24 brd 140.112.XXX.XXX scope global dynamic enp0s3
  valid_lft 82258sec preferred_lft 82258sec
```

- Join a cluster with public IP
  - Even if the node is behind NAT.

```
$ kubeadm join 140.112.XXX.XXX:6443 --token ... --discovery-token-ca-cert-hash sha256:...
```

- Before connecting to NodePort service, find out where is the IP
  - The IP might be public since there is a node over WAN

# For Detailed Information

- Hack MD : https://hackmd.io/@YxfBJOtSQyedaLDrJvmbeA/S17f-X26F

# Forwarding policy research

# Load Balance on Kubernetes

- Service type: Load balancer
- Ingress controller

# Load Balancer

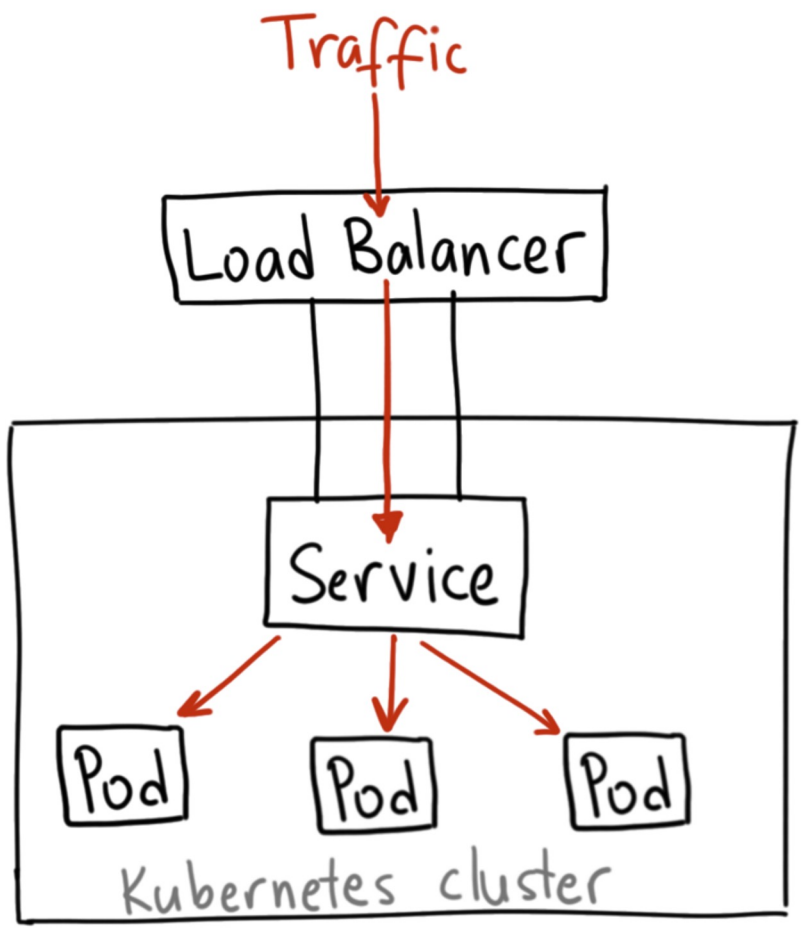
- A Kubernetes service type:

  easy to scale horizontally

- Load balancer is usually provided by cloud provider.

  (GKE, AWS…)

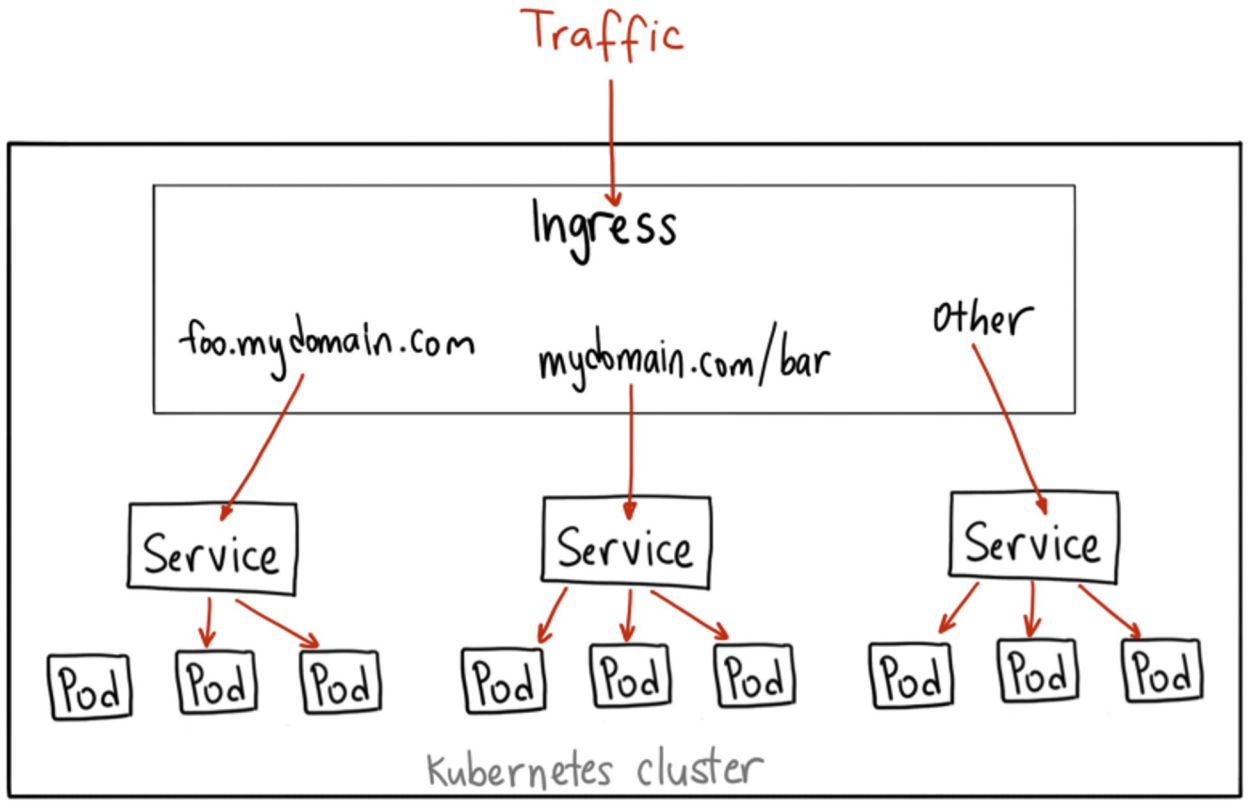
  Hard to manage by ourselves.




pic: https://dockone.io/article/4884

# Ingress

- A Api object that expose outside the cluster.
- Manage and proxy the traffic to services.
- Based on URL resolve.
- Ingress is capable to do load balance.




pic: https://dockone.io/article/4884

# Current problem

- Both Load Balancer services and Ingress are only support static load balancing strategies. For example, round-robin, weighted least request, hash…

- For dynamic load balancing, which base on CPU load, Mem, Network traffic and capacity, is not available on Kubernetes.

# Paper for Current problem

- Solution base on service type load balancer:

  The Design of Multi-Metric Load Balancer for Kubernetes

  Amit Dua; Sahil Randive; Aditi Agarwal; Neeraj Kumar

  https://ieeexplore.ieee.org/author/37085639017

- Solution base on custom ingress:

  Service Dependency Based Dynamic Load Balancing Algorithm for Container Clusters

  Amit Dua; Sahil Randive; Aditi Agarwal; Neeraj Kumar

  https://ieeexplore.ieee.org/author/37088349163

# Current and Future work (1/2)

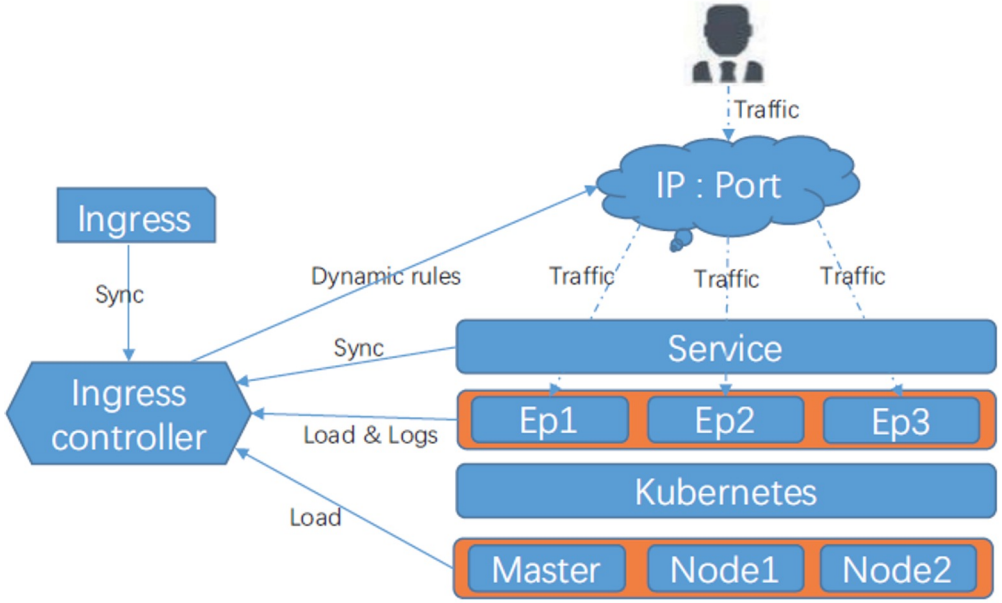
- Decide a solution and implement it on P1935 edge computing platform.

  Paper for service type load balancer is showing how to using promethus and time series database to collect data.

  Paper for custom ingress is more useful because ingress is more convenient to manage a lot of services and this paper also provide a pratical balance algorithm.

# Current and Future work (2/2)

- I am now figuring out a way to combine them since P1935 platform is already using promethus to collect computing resource status.



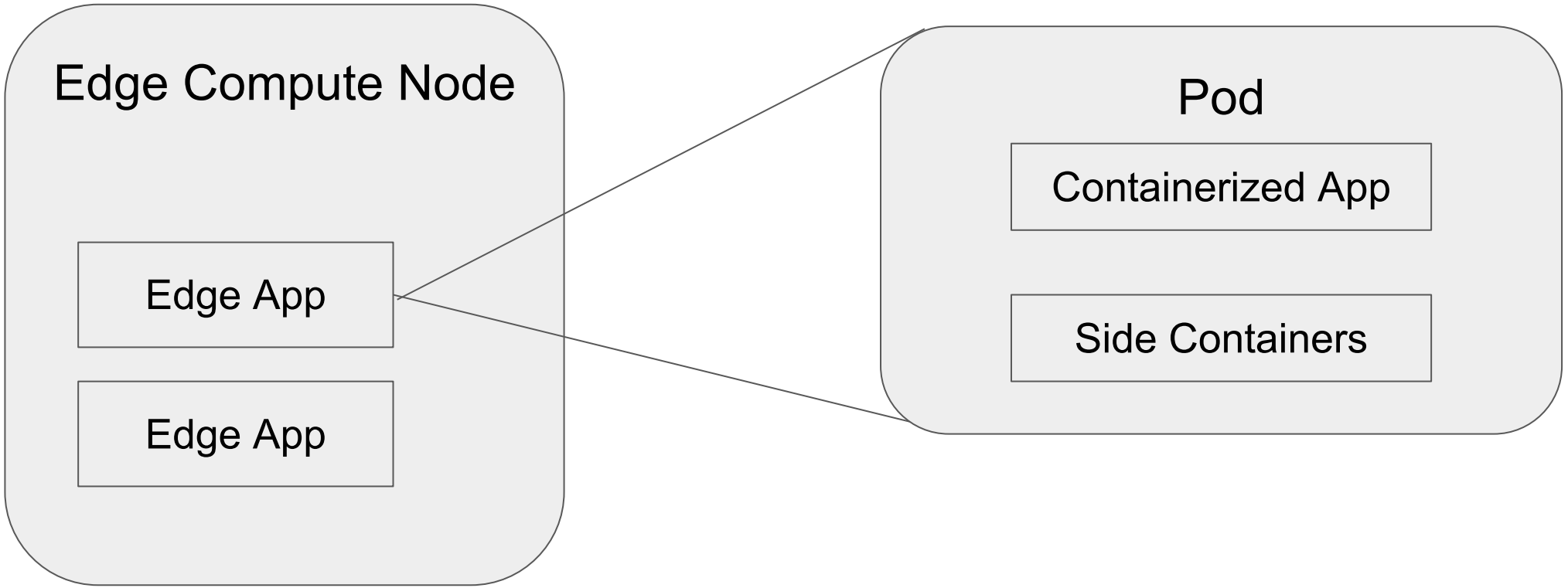

- Find one proper proxy to make sure most of network protocols work.

- Add more traffic balance algorithm.

# Scaling Problem on P1935 Smart Edge System

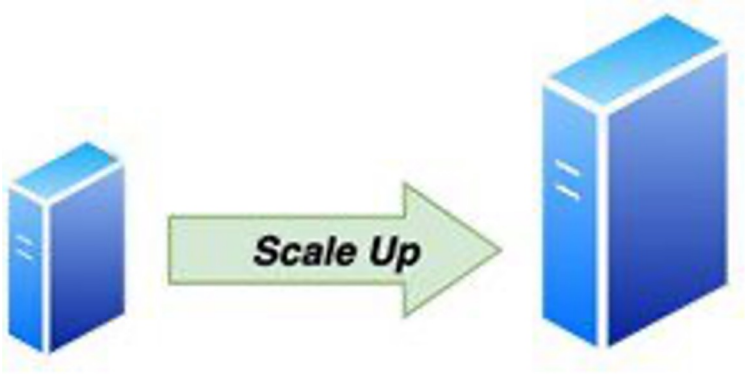
Presenter: 溫進揚

# Introduction



# Scalability of Pods

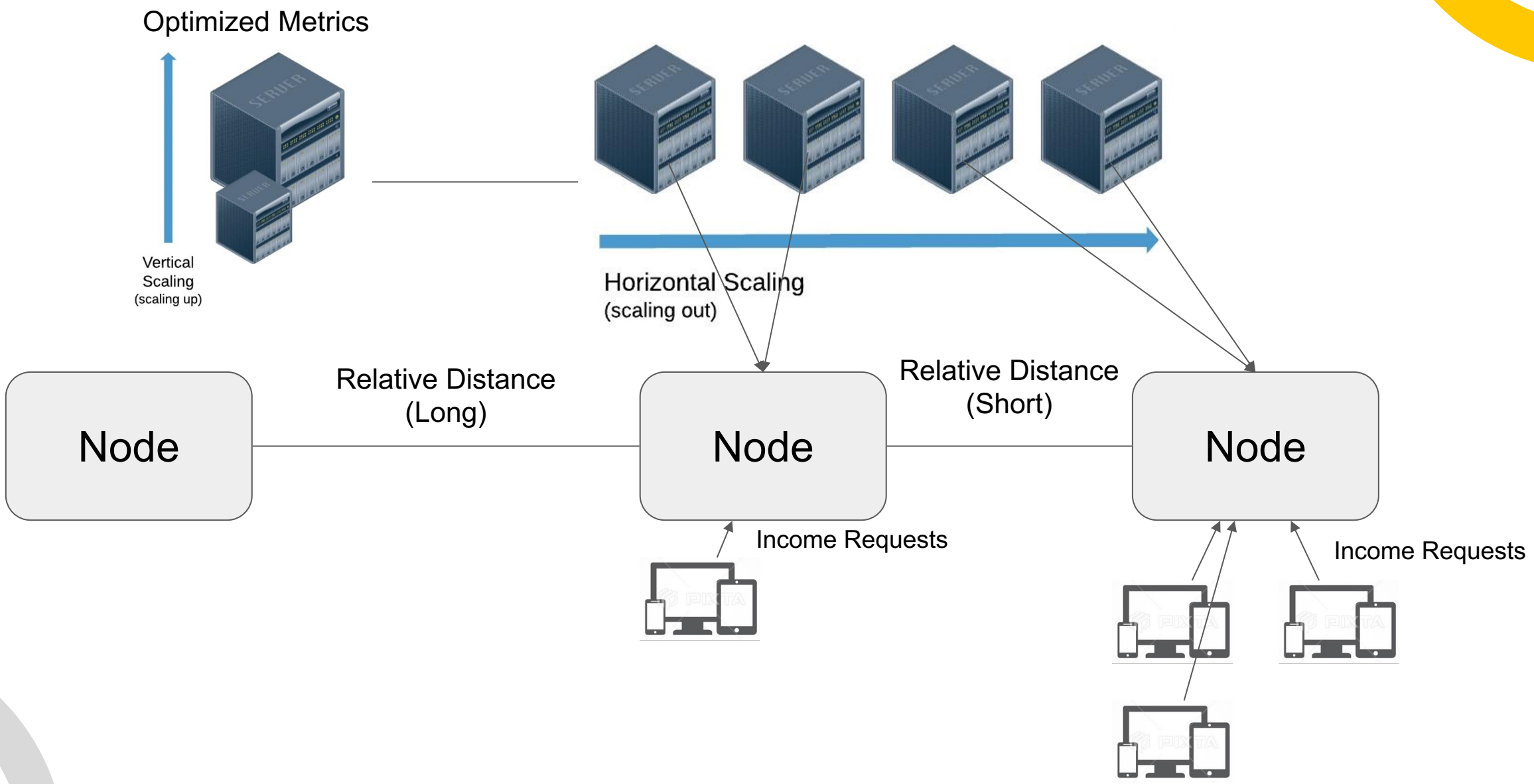
- The ability of a resource or an application to be expanded to handle the increasing demands.
  - Vertical Scaling (VPA)
  - Horizontal Scaling (HPA)




Vertical Scaling

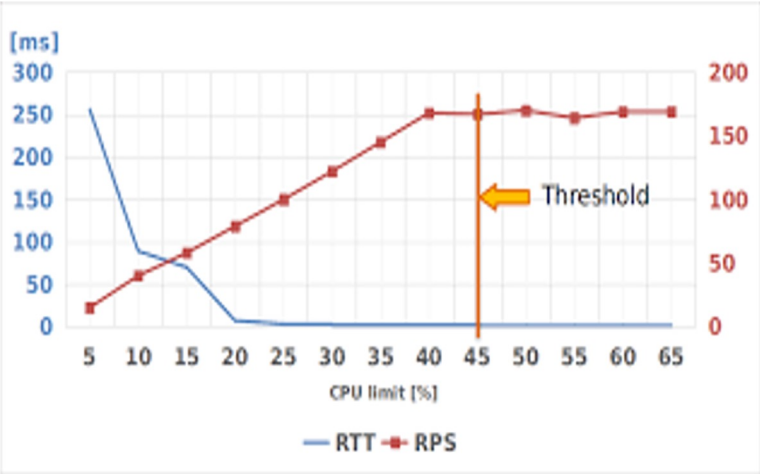

Horizontal Scaling

# Proposal for Pods Scaling and Allocation

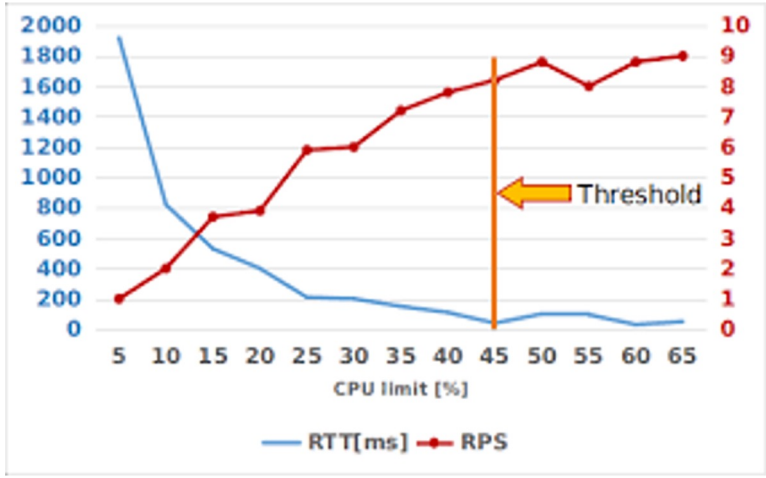


# Adaptive AutoScaling (Libra)

- Automatically detects the optimal resource set for a single pod(VPA), then manages the horizontal scaling process.




Optimized CPU usage for simple web service




Optimized CPU usage for random sized matrix multiplication

Ref: Adaptive scaling of Kubernetes Pods, David Balla

# Traffic aware HPA

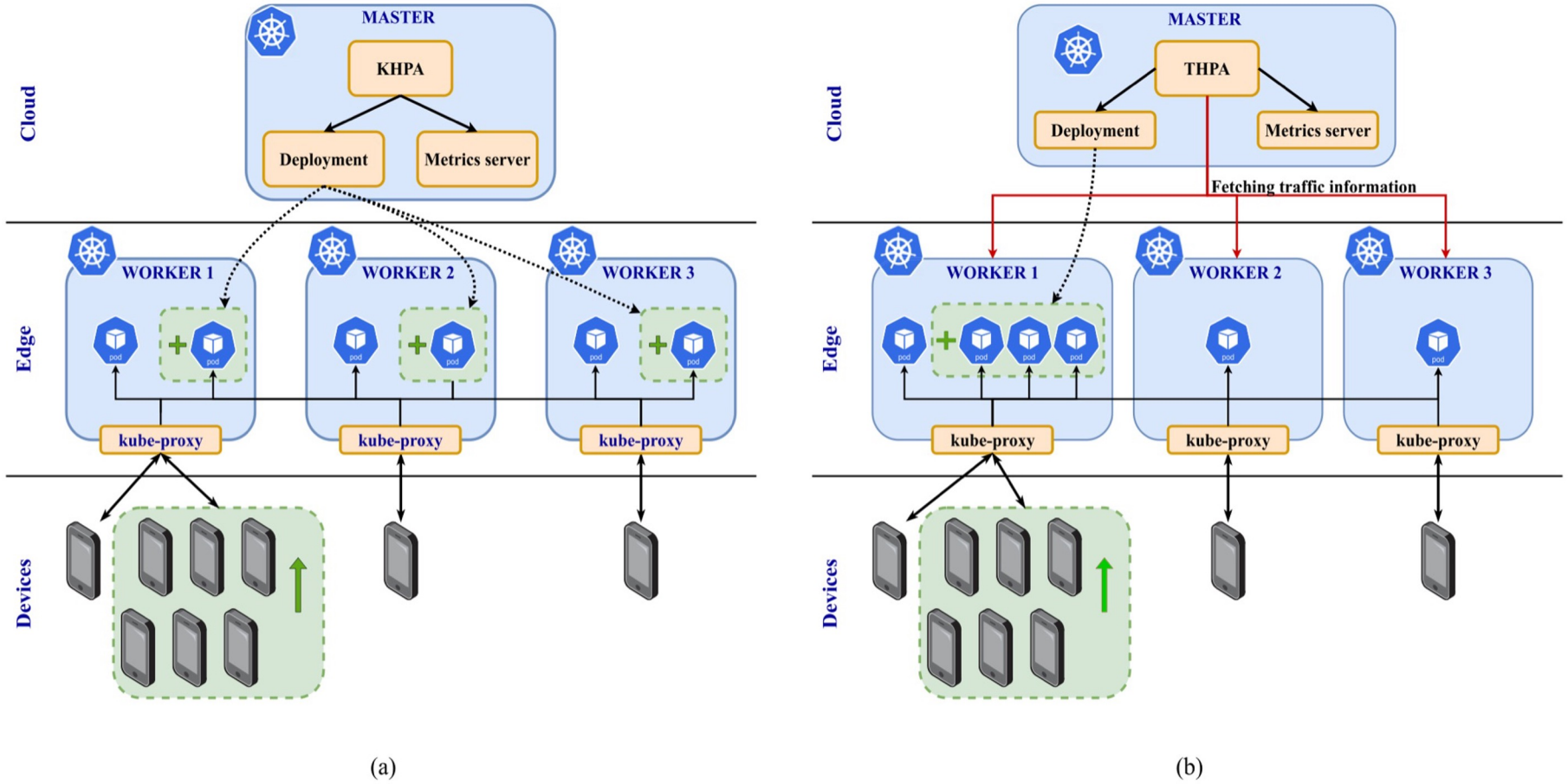



**FIGURE 1.** (a) KHPA in Kubernetes-based Edge computing Architecture and (b) THPA in Kubernetes-based Edge computing Architecture.

Ref: Traffic Aware Horizontal Pod Autoscaler in Kubernetes-Based Edge Computing Infrastructure, LE HOANG PHUC

# Comparison of THPA and KHPA

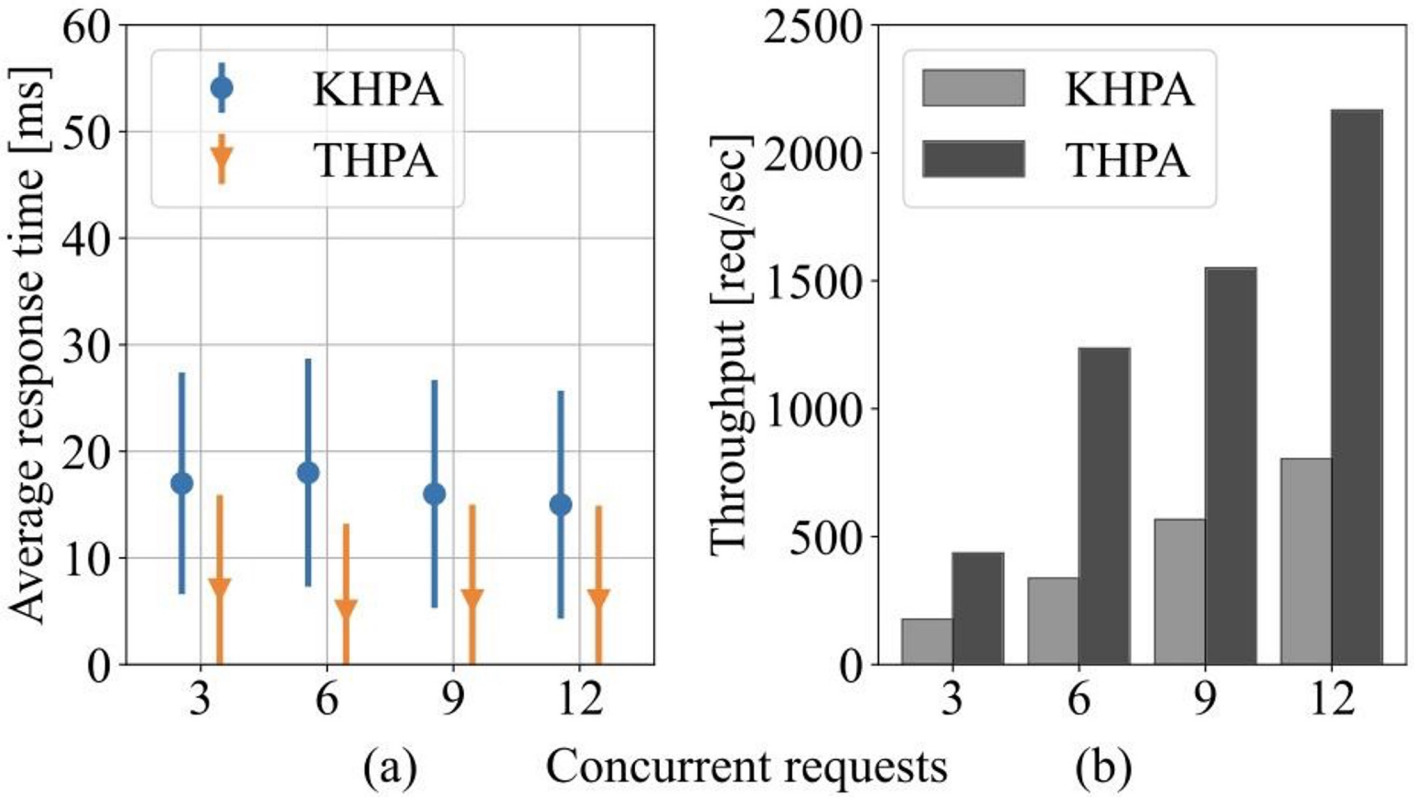
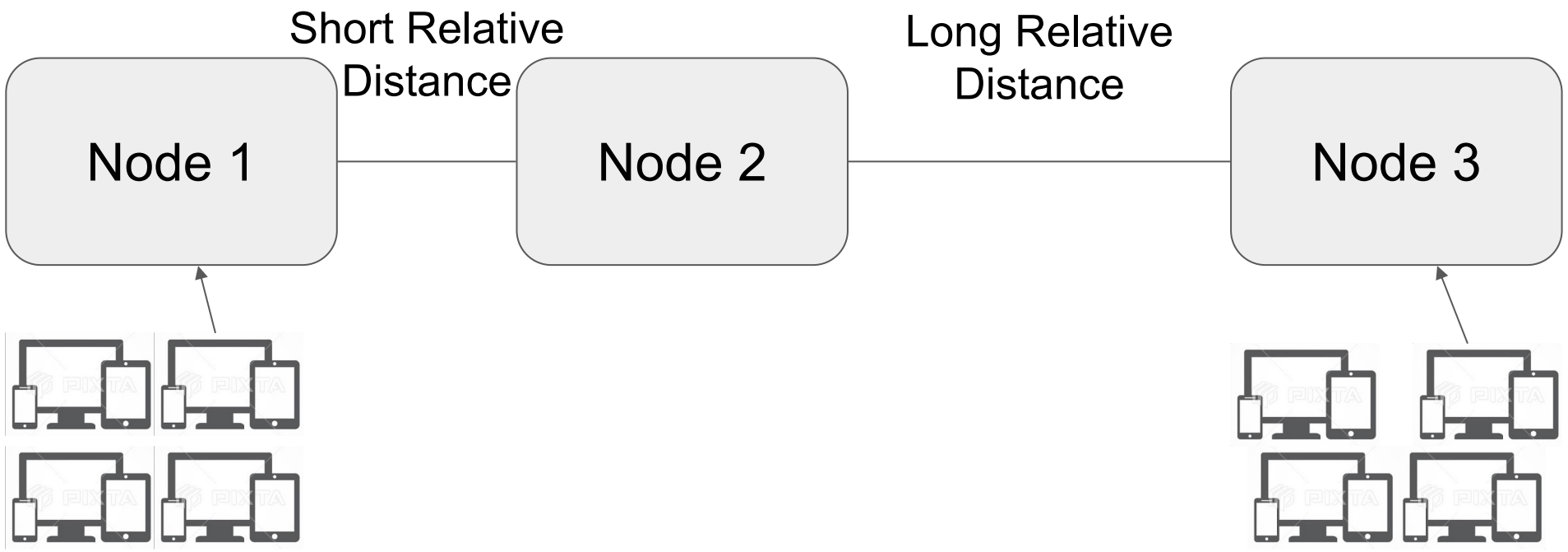


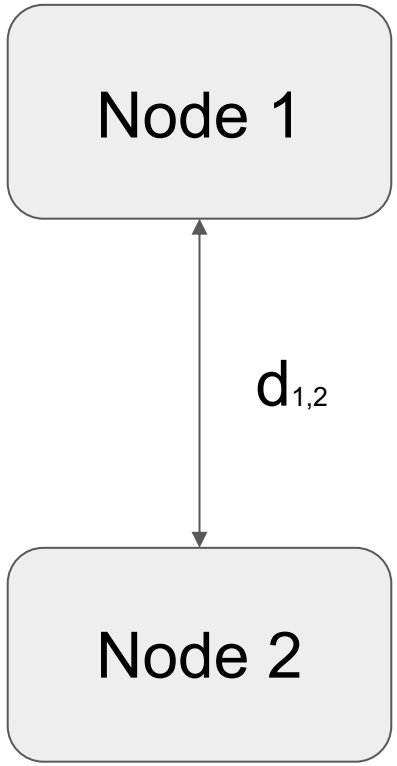

**FIGURE 4. Application performance at worker 1 (a) Response time, and (b) Throughput.**

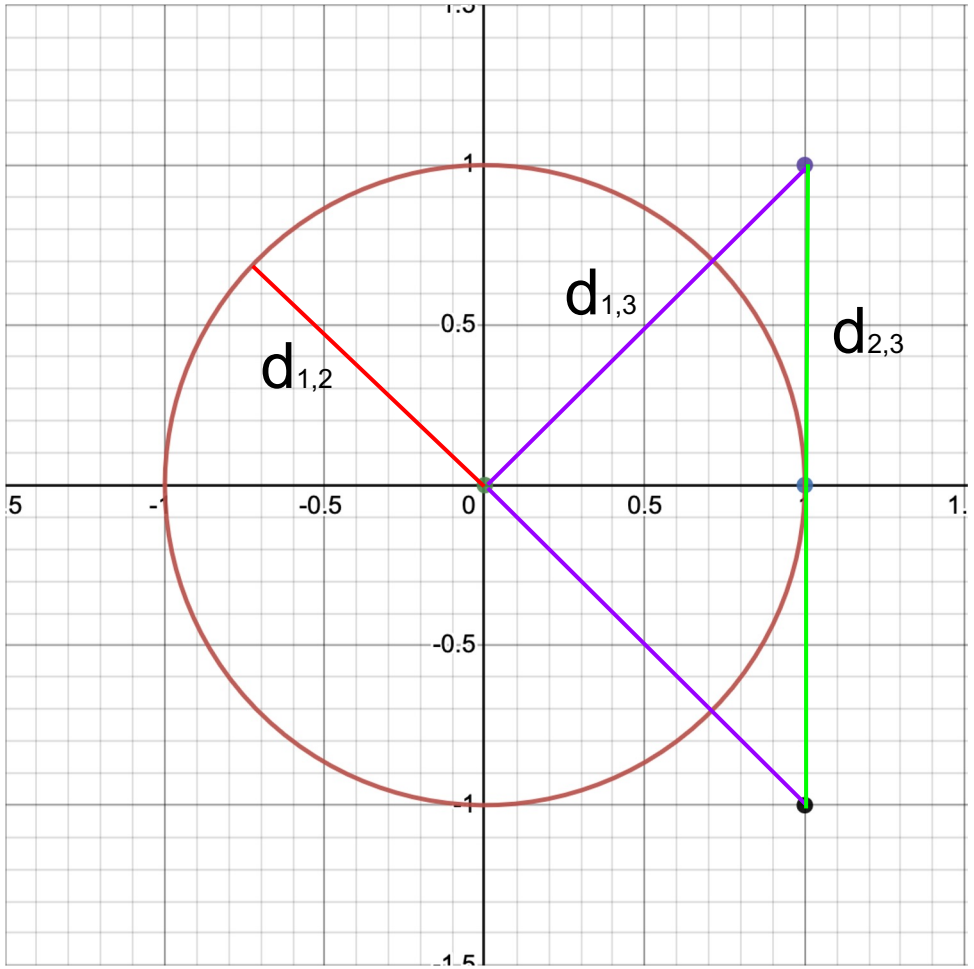
Ref: Traffic Aware Horizontal Pod Autoscaler in Kubernetes-Based Edge Computing Infrastructure, LE HOANG PHUC

# Relative Distance between Nodes



# Relative Location Estimation



# Summary

- We proposed a method to autoscale kubernetes pods which satisfy the usage of edge system.

- The proposed method is both traffic and distance-aware and adaptive so that it may suitable for the dynamic nature of the load in the edge system.

- The proposed method can be further improved in various aspects.
  - Other metric on VPA’s target optimization
  - Infrastructure’s condition aware THPA

# THANKS

## Do you have any questions?